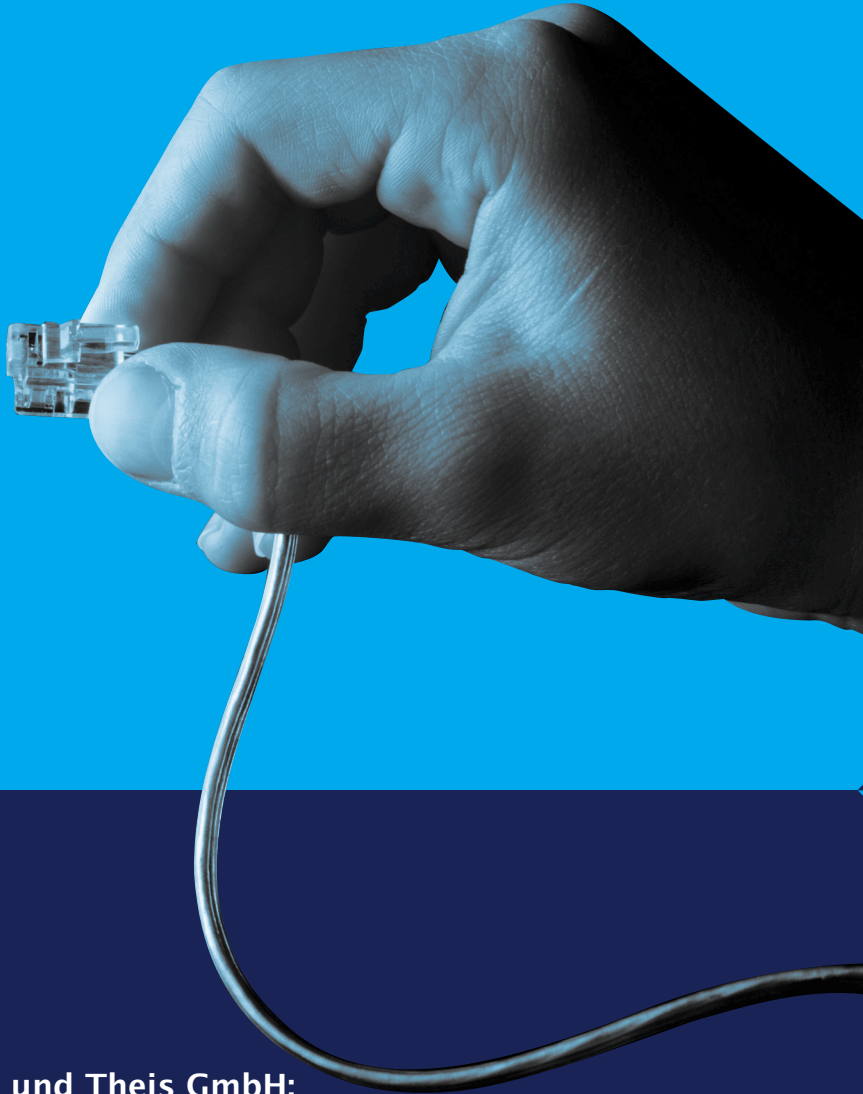


Frank Thiel
Rüdiger Theis

TCP/IP-Ethernet und Web-IO

4. überarbeitete Auflage



Wiesemann und Theis GmbH:
Technische Grundlagen
konkret, kompakt, verständlich

W&T
www.wut.de



W&T

www.wut.de

Wiesemann & Theis GmbH

Technische Grundlagen

konkret

kompakt

verständlich

www.wut.de



TCP/IP-Ethernet und Web-IO

Dieses Heft ist für alle gemacht, die ohne Spezialwissen über Computernetzwerke Ethernet-Endgeräte unter TCP/IP in Betrieb nehmen wollen. Es ist in vier Teile gegliedert:

- **TCP/IP-Ethernet verstehen**

Hier finden Sie die wichtigsten Grundlageninformationen zum Thema TCP/IP.

- **Weitere Protokolle und Dienste**

In diesem Abschnitt erfahren Sie, wie E-Mail funktioniert, was beim Aufruf einer Webseite geschieht, und welche wichtigen Protokolle und Dienste Ihnen im Zusammenhang mit TCP/IP-Ethernet sonst noch begegnen können.

- **TCP/IP -Ethernet Einrichten**

Hier wird Schritt für Schritt die Einrichtung von TCP/IP-Ethernet auf PCs mit den gängigen Betriebssystemen aufgezeigt.

- **Kleines Netzwerk-ABC**

Hier erläutern wir die wichtigsten Begriffe und Abkürzungen, die Ihnen beim Umgang mit Netzwerken begegnen können.

Alle wichtigen Abläufe und Zusammenhänge werden leicht verständlich erklärt.

Keine Angst: Wir werden uns dort nicht bis ins letzte Detail verlaufen. Wir haben uns ganz bewusst auf die Dinge beschränkt, die zum Verständnis der beschriebenen Technologien wirklich wichtig sind.

Zur reinen Inbetriebnahme von TCP/IP-Netzwerkkomponenten ist es schließlich auch gar nicht nötig, jedes Protokoll bis ins letzte Bit zu kennen.

Autoren:

Frank Thiel und Rüdiger Theis

© 01/2003 by Wiesemann & Theis GmbH

4. überarbeitete Auflage

50.001 - 80.000

Nachdruck, auch auszugsweise, mit Quellenangabe einschließlich Internetadresse von W&T (<http://www.wut.de>) ausdrücklich erlaubt.

Microsoft, MS-DOS, Windows, Winsock und Visual Basic sind eingetragene Warenzeichen der Microsoft Corporation.

Irrtümer und Änderungen vorbehalten.

Da wir Fehler machen können, darf keine unserer Aussagen ungeprüft verwendet werden. Bitte melden Sie uns alle Ihnen bekannt gewordenen Irrtümer oder Mißverständlichkeiten, damit wir diese so schnell wie möglich erkennen und beseitigen können.

Inhalt

| | |
|---|-----------|
| Vorwort | 1 |
| TCP/IP-Ethernet verstehen | 7 |
| Anforderungen an ein Computernetzwerk | 8 |
| Grundsätzliche Funktion von Netzwerken | 10 |
| Ethernet und FastEthernet | 11 |
| 10Base2 | 11 |
| 10BaseT | 11 |
| 10Base 5 | 12 |
| 100Base T4 | 12 |
| 100BaseTX | 12 |
| TCP/IP – die wichtigsten Protokolle | 15 |
| IP – Internet Protocol | 15 |
| IP-Adressen | 16 |
| Class A | 16 |
| Class B | 16 |
| Class C | 17 |
| IP-Datenpakete | 18 |
| TCP – Transport Control Protocol | 19 |
| UDP – User Datagramm Protocol | 22 |
| TCP/IP-Ethernet | 23 |
| ARP – Address Resolution Protocol | 26 |
| Gateway und Subnet-Mask | 28 |
| Netzübergreifende TCP/IP-Verbindung | 31 |
| DHCP – Dynamic Host Configuration Protocol | 36 |
| Vergabe der IP-Adresse aus einem Adresspool | 37 |
| Vergabe einer reservierten IP-Adresse | 38 |
| Ausschluss bestimmter IP-Adressen | 40 |
| aus der DHCP-Konfiguration | 40 |
| DHCP und Router | 40 |
| DNS – das Domain Name System | 41 |
| Domainnamen | 41 |
| Namensauflösung im DNS | 43 |
| DNS in Embedded-Systemen | 44 |
| DHCP und DNS | 45 |

Weitere Protokolle und Dienste 47**WWW – World Wide Web48**

URL - Uniform Resource Locator 49

HTML – Hypertext Markup Language 52

Grundsätzlicher Aufbau einer HTML-Datei 53

Hyperlinks 54

Darstellung von multimedialen Inhalten 55

HTTP – Hypertext Transfer Protocol 59

Die wichtigsten HTTP-Kommandos und Parameter 60

Das GET-Kommando 60

Das POST-Kommando 62

Das HEAD-Kommando 63

HTTP-Versionen 63

Interaktivität im WWW65

Interaktivität durch Programme die auf dem Server ablaufen .. 65

CGI - Common Gateway Interface 65

PHP 66

Programme die im Browser ausgeführt werden. 67

JavaScript 67

Java Applets 69

E-Mail71

Aufbau einer E-Mail 72

MIME – Multipurpose Internet Mail Extensions 74

SMTP – Simple Mail Transfer Protocol 74

POP3 – Post Office Protocol Version 3 75

E-Mail über HTTP senden und empfangen 76

E-Mail und DNS 78

Telnet - Terminal over Network80

Der Telnet Client 80

Der Telnet-Server 81

Das Telnet Protokoll 81

| | |
|--|----------------|
| FTP - File Transfer Protocol | 84 |
| Der FTP-Client | 84 |
| Das FTP-Protokoll | 85 |
| Der FTP-Server | 87 |
| TFTP - Trivial File Transfer Protocol | 88 |
| SNMP - Simple Network Management Protocol | 92 |
| Modbus-TCP | 93 |
| Socket-Programmierung | 94 |
| TCP-Client, TCP-Server oder UDP-Peer? | 95 |
| TCP-Client | 96 |
| TCP-Server | 96 |
| UDP | 97 |
| Socket-Programmierung in Visual Basic | 98 |
| Ein TCP-Client in VB | 98 |
| Ein TCP-Server in VB | 103 |
| Ein einfacher UDP-Peer in VB | 107 |
| Socket-Programmierung in Delphi | 109 |
| Ein TCP-Client in Delphi | 109 |
| Ein TCP-Server in Delphi | 114 |
| TCP/IP -Ethernet Einrichten | 121 |
| TCP/IP unter Windows 9x installieren und konfigurieren | 122 |
| TCP/IP unter Windows NT installieren und konfigurieren | 126 |
| TCP/IP unter Win 2000 installieren und konfigurieren | 129 |
| TCP/IP-Ethernet bei gleichzeitigem DFÜ-Internetzugang .. | 131 |
| Kleines Netzwerk-ABC | 134 |
| Zahlensysteme | 150 |

| | |
|---|------------|
| Web-IO | 151 |
| Com-Server - Anwendungsbeispiele aus der Praxis | 152 |
| Box-to-Box - Der Tunnel durchs Netzwerk | 153 |
| Die COM-Umlenkung - Der „ganz wo anders“ COM-Port | 154 |
| TCP/IP-Sockets - Mit dem eigenen Programm auf den seriellen Port | 155 |
| FTP - Serielle Daten direkt in eine Datei | 156 |
| Com-Server - Die verschiedenen Modelle | 157 |
| Com-Server Highspeed Industry - #58631 | 157 |
| Com-Server Highspeed - #58031, 58034 | 158 |
| OEM Platinen | 158 |
| Web-IO - Anschlussbeispiele aus der Praxis | 159 |
| Web-IO Thermometer - Temperaturüberwachung im Netz ... | 160 |
| Web-IO 12xDigital | 162 |
| Web-IO - verschiedene Modelle | 164 |
| Web-IO Thermometer #57603, 57604 | 164 |
| Web-IO 12xDigital #57630, 58631 | 165 |
| Weitere Infos | 166 |

TCP/IP-Ethernet verstehen

Noch vor wenigen Jahren waren Computernetzwerke nur in Banken, Behörden und größeren Betrieben zu finden. Die verwendeten Netzwerkkomponenten waren meist kaum zu bezahlen, Installation und Administration ließen sich nur von speziell ausgebildeten Fachleuten bewältigen.

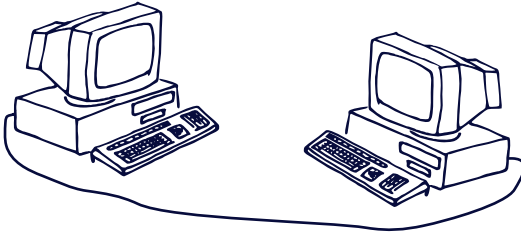
Doch spätestens seit den 90er Jahren nahmen Computer – vor allem der PC – rasanten Einzug in alle Bereiche des täglichen Lebens; ein immer höheres Datenaufkommen trug wesentlich zur Verbreitung und verstärkten Nutzung von Computernetzwerken bei.

Parallel zu dieser Entwicklung breitete sich das Internet explosionsartig aus und kann heute auch von privaten Anwendern problemlos in Anspruch genommen werden.

Dies alles hat dazu geführt, dass die Möglichkeit zum Zugriff auf Computernetzwerke heute fester Bestandteil moderner Betriebssysteme ist. Die wichtigste Rolle kommt dabei zwei Dingen zu: Ethernet als physikalischer Grundlage und TCP/IP als Protokoll.

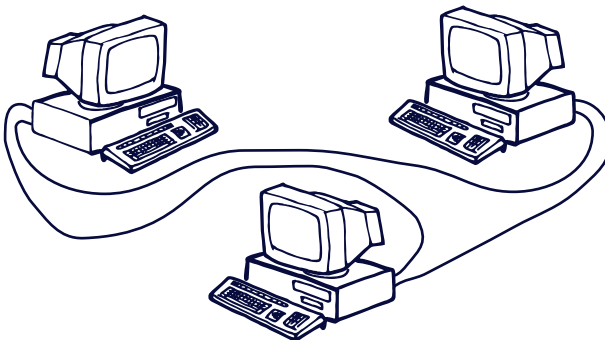
Anforderungen an ein Computernetzwerk

Jeder Computerbenutzer hat sicher schon einmal zwei Endgeräte (z.B. PC und Drucker, PC und Modem, PC und PC) miteinander verbunden. Zur Verbindung dient ein für den speziellen Anwendungsfall vorgesehenes Kabel, über das Daten zwischen beiden Geräten hin und her geschickt werden.



Man kann sich das so vorstellen: Zwei Brieffreunde senden einander Briefe, und ein Bote ist ständig damit beschäftigt, diese Briefe zu den Briefkästen der beiden zu befördern. In diesem einfachen Fall sind weder Briefumschlag noch Adresse oder Absender nötig.

Das Verfahren ist unkompliziert und funktioniert reibungslos. Es werden nur die reinen Nutzdaten verschickt. Diese Art der Verbindung nennt man auch Punkt-zu-Punkt-Verbindung. Man könnte die Punkt-zu-Punkt-Verbindung natürlich auch nutzen, um z.B. drei PCs miteinander kommunizieren zu lassen. Dazu müsste also von jedem PC je ein Kabel zu den beiden anderen PCs verlegt werden.



Für den Versand von Briefen zwischen drei Brieffreunden würden bei diesem Verfahren drei Boten gebraucht.

Schon bei vier beteiligten PCs brauchte man aber sechs Kabel, und wenn man zehn oder mehr PCs auf diese Weise „vernetzen“ wollte, wäre ein unentwirrbarer Kabelknoten die Folge. Außerdem würde jede Veränderung eines solchen Netzwerkes eine ganze Lawine von Änderungen in der Verkabelung nach sich ziehen. Die Umsetzung einer solchen Vernetzung ist also wenig praktikabel.

Ein Computernetzwerk sollte bei geringstem Material- und Verkabelungsaufwand, vorhandene Ressourcen (Speicherplatz, Datenbanken, Drucker und andere beliebige Endgeräte) einer unbestimmten Zahl von angeschlossenen Nutzern zugänglich machen. Dabei muss ein Höchstmaß an Datensicherheit und Übertragungsgeschwindigkeit gegeben sein.

Aus diesen Anforderungen heraus entstanden die heute üblichen Netzwerkstandards.

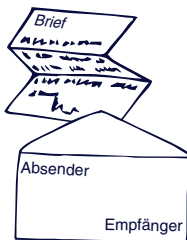
Grundsätzliche Funktion von Netzwerken

Grundsätzlich haben alle Netzwerktopologien eines gemeinsam:

Jeder Netzteilnehmer erhält eine eigene Adresse. Die Nutzdaten werden in einem Rahmen aus zusätzlichen Informationen (z. B. Adresse des Empfängers, Adresse des Absenders und Checksumme) „eingepackt“.

Mit Hilfe der Adressinformationen, in den so entstandenen Datenpaketen, können die Nutzdaten über gemeinsam benutzte Leitungswege an den richtigen Empfänger übermittelt werden.

Bei einem Brief ist es nicht anders: Man steckt den Brief in einen Umschlag, auf dem Empfänger und Absender notiert sind. Der Postbote weiß dann, wem er den Brief zustellen soll; der Empfänger kann ablesen, woher er kommt und wem er bei Bedarf zu antworten hat.



Beim Datentransfer innerhalb eines Netzwerkes hat der Empfänger zusätzlich die Möglichkeit, mit Hilfe der mitversandten Checksumme die Vollständigkeit der empfangenen Nutzdaten zu überprüfen.

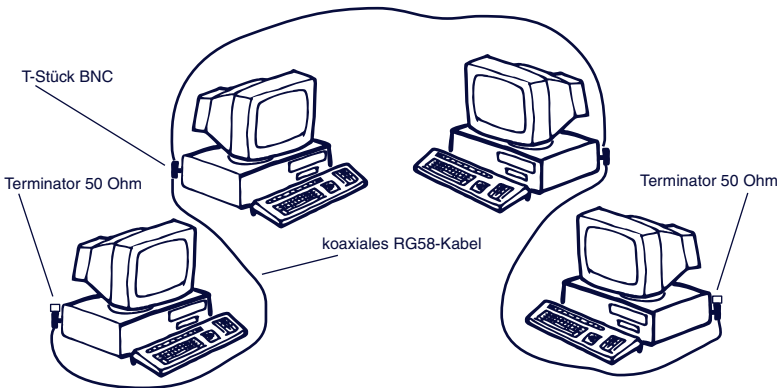
Ethernet und FastEthernet

Ethernet ist der heute am meisten verbreitete Netzwerkstandard. Bereits 1996 waren ca. 86% aller bestehenden Netzwerke in dieser Technologie realisiert.

Ethernet wurde ursprünglich mit einer Übertragungsgeschwindigkeit von 10Mbit/s betrieben; hierbei gibt es drei verschiedene physikalische Grundmodelle:

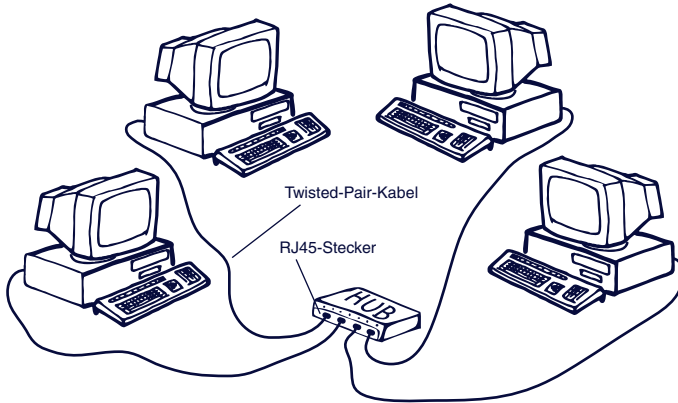
10Base2

Auch bekannt als Thin Ethernet, Cheapernet oder schlicht als BNC-Netzwerk. Alle Netzteilnehmer werden parallel auf ein Koaxialkabel (RG58, 50 Ohm Wellenwiderstand) aufgeschaltet. Das Kabel muß an beiden Seiten mit einem 50-Ohm-Terminator (Endwiderstand) abgeschlossen sein.



10BaseT

Jeder Netzteilnehmer wird über ein eigenes Twisted-Pair-Kabel an einen sogenannten Hub (Sternverteiler) angeschlossen, der alle Datenpakete gleichermaßen an alle Netzteilnehmer weitergibt. 10BaseT arbeitet also physikalisch sternförmig, aber logisch – wie auch 10Base2 – nach dem Busprinzip.



10Base 5

Auch oft als „Yellow Cable“ bezeichnet; stellt den ursprünglichen Ethernetstandard dar und hat heute kaum noch Bedeutung.

Mit zunehmend größeren Datenmengen wurde in den 90er Jahren Fast Ethernet mit einer Übertragungsgeschwindigkeit von 100Mbit/s eingeführt; hierbei gibt es zwei verschiedene physikalische Grundmodelle:

100Base T4

Genau wie bei 10BaseT wird jeder Netzteilnehmer über ein eigenes Twisted-Pair-Kabel an einen Hub angeschlossen, der alle Datenpakete an alle Netzteilnehmer weitergibt. 100BaseT4 kommt bei Neuinstallationen kaum noch zur Anwendung.

100BaseTX

stellt den heute üblichen Standard für 100Mbit Netzwerke dar. 100BaseT4 und 100BaseTX unterscheiden sich nur auf physikalischer Ebene in der Art der Datenübertragung. Außerdem benötigt 100BaseTX eine höherwertige Verkabelung.

Ausführlichere Spezifikationen zu Ethernet und den verschiedenen physikalischen Topologien finden Sie auch im W&T DatenBuch.

Welches physikalische Grundmodell auch genutzt wird – der logische Aufbau der verwendeten Datenpakete ist bei allen Ethernet-Topologien gleich. Alle Netzteilnehmer in einem lokalen Netz erhalten alle Datenpakete einschließlich derer, die für die anderen Netzteilnehmer bestimmt sind (zur Ausnahme *Switch* vgl. den Anhang), verarbeiten aber nur diejenigen Pakete weiter, die tatsächlich an sie selbst adressiert sind.

Die Ethernetadresse – auch MAC-ID oder Node-Number genannt – wird vom Hersteller in den physikalischen Ethernetadapter (Netzwerkkarte, Printserver, Com-Server, Router ...) fest „eingeschnitten“, steht also für jedes Endgerät fest und kann nicht geändert werden. Die Ethernet-Adresse ist ein 6-Byte-Wert, der üblicherweise in hexadezimaler Schreibweise angegeben wird. Eine Ethernetadresse sieht typischerweise so aus: 00-C0-3D-00-27-8B.



*Jede Ethernet-Adresse
ist weltweit einmalig!*

Die ersten drei Hex-Werte bezeichnen dabei den Herstellercode, die letzten drei Hex-Werte werden vom Hersteller fortlaufend vergeben.


Es gibt vier verschiedene Typen von Ethernet-Datenpaketen, die je nach Anwendung eingesetzt werden:

| <i>Datenpakettyp</i> | <i>Anwendung</i> |
|----------------------|----------------------------|
| Ethernet 802.2 | Novell IPX/SPX |
| Ethernet 802.3 | Novell IPX/SPX |
| Ethernet SNAP | APPLE TALK Phase II |
| Ethernet II | APPLE TALK Phase I, TCP/IP |

In Verbindung mit TCP/IP werden in aller Regel Ethernet-Datenpakete vom Typ Ethernet II verwendet.

Hier der Aufbau eines Ethernet-II-Datenpakets:

Aufbau eines Ethernet-Datenpakets

| | | | | | |
|--|--------------|--------------|------|-----------|-------------|
|  | 00C03D00278B | 03A055236544 | 0800 | Nutzdaten | Check-summe |
| Preamble | Destination | Source | Type | Data | FCS |

Preamble Die Bitfolge mit stetigem Wechsel zwischen 0 und 1 dient zur Erkennung des Paketanfangs bzw. der Synchronisation. Das Ende der Preamble wird durch die Bitfolge „11“ gekennzeichnet.

Destination Ethernet-Adresse des Empfängers.

Source Ethernet-Adresse des Absenders.

Type Gibt den übergeordneten Verwendungszweck an (z.B. IP = Internet Protocol = 0800h).

Data Nutzdaten.

FCS Checksumme.

Der Aufbau der anderen Ethernet-Pakete unterscheidet sich nur in den Feldern *Type* und *Data*, denen je nach Pakettyp eine andere Funktion zukommt. Damit verfügt ein Ethernet-Datenpaket über sämtliche erforderlichen Eigenschaften, um in lokalen Netzwerken Daten von einem Netzteilnehmer zum anderen zu verschicken.

Ethernet allein verfügt allerdings nicht über die Möglichkeit, verschiedene Netze zu adressieren. Darüber hinaus arbeitet Ethernet verbindungslos: Der Absender erhält vom Empfänger keine Bestätigung, ob ein Paket angekommen ist.

Spätestens wenn ein Ethernet-Netzwerk mit mehreren Netzen verbunden werden soll, muss also mit übergeordneten Protokollen – etwa mit TCP/IP – gearbeitet werden.

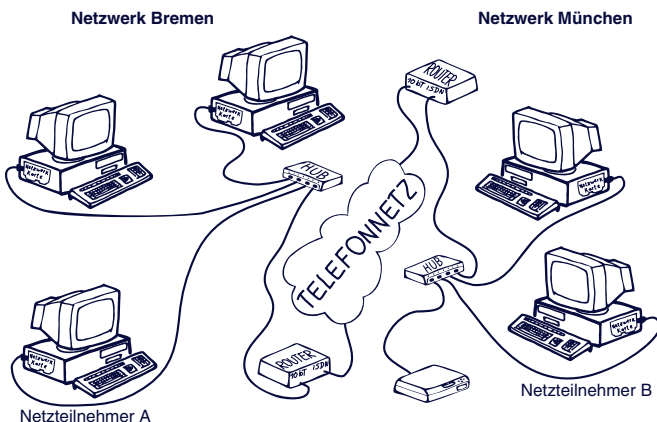
TCP/IP – die wichtigsten Protokolle

Bereits in den 60er Jahren vergab das amerikanische Militär den Auftrag, ein Protokoll zu schaffen, das unabhängig von der verwendeten Hard- und Software einen standardisierten Informationsaustausch zwischen einer beliebigen Zahl verschiedener Netzwerke möglich machen sollte. Aus dieser Vorgabe entstand im Jahr 1974 das Protokoll TCP/IP.

Obwohl TCP und IP immer in einem Wort genannt werden, handelt es sich hier um zwei aufeinander aufsetzende Protokolle. Das Internet Protocol IP übernimmt die richtige Adressierung und Zustellung der Datenpakete, während das darauf aufsetzende Transport Control Protocol TCP für den Transport und die Sicherung der Daten zuständig ist.

IP – Internet Protocol

Das Internet-Protokoll macht es möglich, eine unbestimmte Anzahl von Einzelnetzen zu einem Gesamtnetzwerk zusammenzufügen. Es ermöglicht also den Datenaustausch zwischen zwei beliebigen Netzteilnehmern, die jeweils in beliebigen Einzelnetzen positioniert sind. Die physikalische Ausführung der Netze bzw. Übertragungswege (Ethernet, Token Ring, ISDN....) spielt hierbei keine Rolle. Die Daten werden ungeachtet dieser Unterschiede an den Empfänger übermittelt.



IP-Adressen

Unter IP hat jeder Netzteilnehmer eine einmalige Internet-Adresse, die oft auch als „IP-Nummer“ bezeichnet wird. Diese Internet-Adresse ist ein 32-Bit-Wert, der zur besseren Lesbarkeit immer in Form von vier durch Punkte getrennten Dezimalzahlen (8-Bit-Werten) angegeben wird (Dot-Notation).

Die Internet-Adresse unterteilt sich in Net-ID und Host-ID, wobei die Net-ID zur Adressierung des Netzes und die Host-ID zur Adressierung des Netzteilnehmers innerhalb eines Netzes dient.

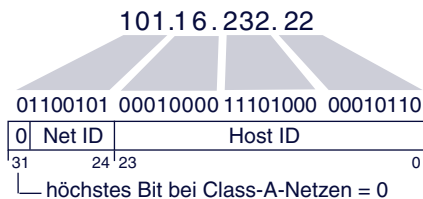
Ähnlich sind auch Telefonnummern aufgebaut. Auch hier unterscheidet man zwischen Vorwahl und Teilnehmer-rufnummer.

Welcher Teil der IP-Adresse zur Net-ID und welcher zur Host-ID gehört, hängt von der Größe des Netzes ab.

Zur Adressierung normaler Netze unterscheidet man drei Netzwerkklassen:

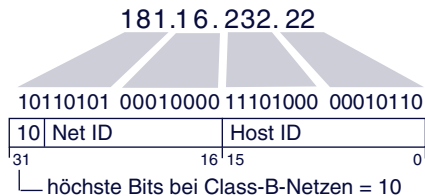
Class A:

Das erste Byte der IP-Adresse dient der Adressierung des Netzes, die letzten drei Byte adressieren den Netzteilnehmer.



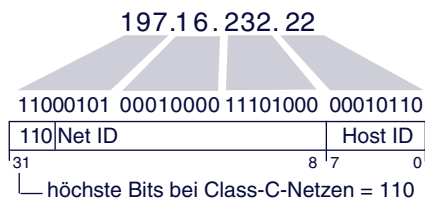
Class B:

Die ersten zwei Byte der IP-Adresse dienen der Adressierung des Netzes, die letzten zwei Byte adressieren den Netzteilnehmer.



Class C:

Die ersten drei Byte der IP-Adresse dienen der Adressierung des Netzes, das letzte Byte adressiert den Netzteilnehmer.



In der folgenden Tabelle finden Sie die Eckdaten der verschiedenen Netzklassen:

| | Adressbereich des Netzwerks | Anzahl möglicher Netze | mögl. Anzahl Hosts/Netz |
|---------|---------------------------------|------------------------------|-------------------------------|
| Class A | 1.xxx.xxx.xxx – 126.xxx.xxx.xxx | 127 (2^7) | ca. 16 Millionen (2^{24}) |
| Class B | 128.0.xxx.xxx – 191.255.xxx.xxx | ca. 16.000 (2^{14}) | ca. 65.000 (2^{16}) |
| Class C | 192.0.0.xxx – 223.255.255.xxx | ca. 2 Millionen (2^{21}) | 254 (2^8) |

Neben den hier aufgeführten Netzen, gibt es auch noch Class-D- und Class-E-Netze, deren Adressbereiche oberhalb der Class-C-Netze liegen. Class-D-Netze und Class-E-Netze haben in der Praxis wenig Bedeutung, da sie nur zu Forschungszwecken und für Sonderaufgaben verwendet werden. Der normale Internetbenutzer kommt mit diesen Netzwerkklassen nicht in Berührung.

Für Netzwerke, die direkt mit dem Internet verbunden werden sollen, vergibt eine Kommission namens InterNIC eine freie Net-ID und entscheidet abhängig von der geplanten Netzgröße, um welche Netzklasse es sich handelt.

Die Zuordnung der Host-ID zum Netzteilnehmer und damit die daraus resultierende IP-Adresse kann der Netzbetreiber (Admi-

nistrator) frei wählen. Er muss dabei allerdings Sorge tragen, dass eine IP-Adresse zu einer Zeit nur einmal vergeben sein darf.

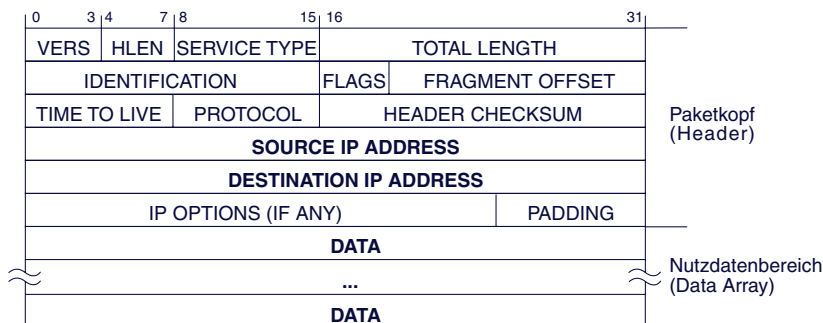


Eine IP-Adresse muß im gesamten verbundenen Netzwerk einmalig sein!

IP-Datenpakete

Auch bei der Datenübertragung über das Internet werden die Nutzdaten in einen Rahmen von Adressierungsinformationen gepackt. IP-Datenpakete enthalten neben den zu transportierenden Nutzdaten eine Fülle von Adress- und Zusatzinformationen, die im sogenannten „Paketkopf“ stehen.

Aufbau eines IP-Datenpakets



Wir beschränken uns hier auf die Erklärung der wichtigsten Adressinformationen:

source IP address: IP-Adresse des Absenders

destination IP address: IP-Adresse des Empfängers

TCP – Transport Control Protocol

Weil IP ein ungesichertes, verbindungsloses Protokoll ist, arbeitet es in der Regel mit dem aufgesetzten TCP zusammen, das die Sicherung und das Handling der Nutzdaten übernimmt.

TCP stellt für die Dauer der Datenübertragung eine Verbindung zwischen zwei Netzteilnehmern her. Beim Verbindungsaufbau werden Bedingungen wie z.B. die Größe der Datenpakete festgelegt, die für die gesamte Verbindungsdauer gelten.

TCP kann man mit einer Telefonverbindung vergleichen. Teilnehmer A wählt Teilnehmer B an; Teilnehmer B akzeptiert mit dem Abheben des Hörers die Verbindung, die dann bestehen bleibt, bis einer der Beiden sie beendet.

TCP arbeitet nach dem sogenannten *Client-Server-Prinzip*:

Denjenigen Netzteilnehmer, der eine Verbindung aufbaut (der also die Initiative ergreift), bezeichnet man als Client. Der Client nimmt einen vom Server angebotenen Dienst in Anspruch, wobei je nach Dienst ein Server auch mehrere Clients gleichzeitig bedienen kann.

Derjenige Netzteilnehmer, zu dem die Verbindung aufgebaut wird, wird als Server bezeichnet. Ein Server tut von sich aus nichts, sondern wartet auf einen Client, der eine Verbindung zu ihm aufbaut.

Im Zusammenhang mit TCP spricht man von TCP-Client und TCP-Server.

TCP sichert die übertragenen Nutzdaten mit einer Checksumme und versieht jedes gesendete Datenpaket mit einer Sequenznummer. Der Empfänger eines TCP-Pakets prüft anhand der Checksumme den korrekten Empfang der Daten. Hat ein TCP-Server ein Paket korrekt empfangen, wird über einen vorgegebenen Algorithmus aus der Sequenznummer eine Acknowledgement-Nummer errechnet.

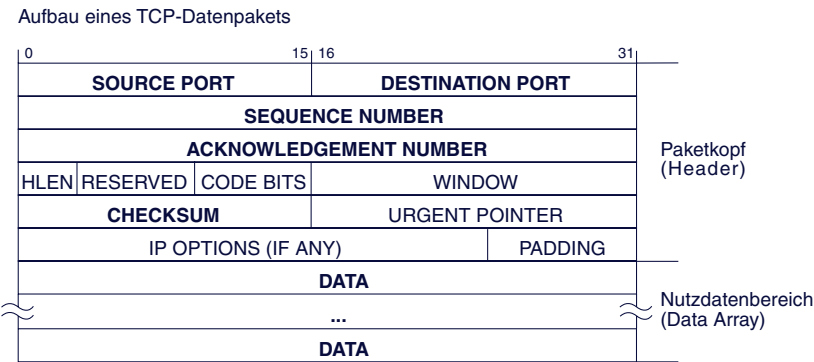
Die Acknowledgement-Nummer wird dem Client mit dem nächsten selbst gesendeten Paket als Quittung zurückgegeben. Der Server versieht seine gesendeten Pakete ebenfalls mit einer eigenen Sequenznummer, die wiederum vom Client mit einer Acknowledgement-Nummer quittiert wird.

Dadurch ist gewährleistet, dass der Verlust von TCP-Paketen bemerkt wird, und diese im Bedarfsfall in korrekter Abfolge erneut gesendet werden können.

Darüber hinaus leitet TCP die Nutzdaten auf dem Zielrechner an das richtige Anwendungsprogramm weiter, indem es unterschiedliche Anwendungsprogramme – auch Dienste genannt – über unterschiedliche Portnummern anspricht. So ist Telnet z.B. über Port 23, FTP über Port 21 zu erreichen.

Vergleicht man ein TCP-Paket mit einem Brief an eine Behörde, kann man die Portnummer mit der Raumnummer der adressierten Dienststelle vergleichen. Befindet sich z.B. das Straßenverkehrsamt in Raum 312 und man adressiert einen Brief an eben diesen Raum, dann gibt man damit zugleich auch an, dass man die Dienste des Straßenverkehrsamts in Anspruch nehmen möchte.

Auch TCP verpackt die Nutzdaten in einen Rahmen von Zusatzinformationen. Solche TCP-Pakete sind wie folgt aufgebaut:

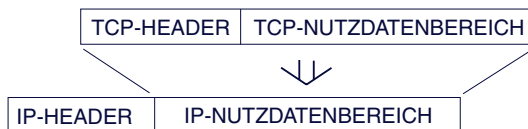


W&T

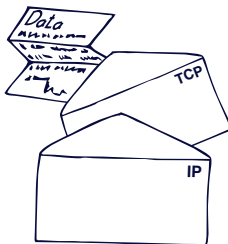
| | |
|--------------------------|--|
| Source Port: | Portnummer der Applikation des Absenders |
| Destination Port: | Portnummer der Applikation des Empfängers |
| Sequence No: | Offset des ersten Datenbytes relativ zum Anfang des TCP-Stroms (garantiert die Einhaltung der Reihenfolge) |
| Acknowl. No: | im nächsten TCP-Paket erwartete Sequence No. |
| Data: | Nutzdaten |

Das so entstandene TCP-Paket wird in den Nutzdatenbereich eines IP-Pakets eingesetzt.

Aufbau eines TCP/IP-Datenpakets



Die Nutzdaten werden quasi in einen Briefumschlag (TCP-Paket) gesteckt, der wiederum in einen Briefumschlag (IP-Paket) gesteckt wird.



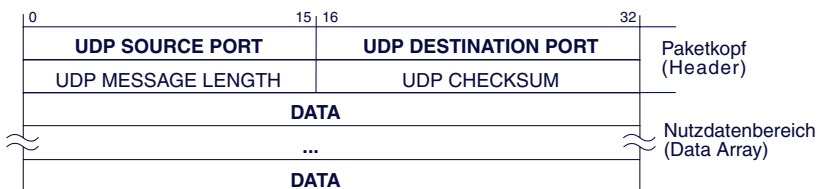
UDP – User Datagramm Protocol

UDP ist ein weiteres Transportprotokoll, das genau wie TCP auf IP aufsetzt. Im Gegensatz zu TCP arbeitet UDP verbindungslos. Jedes Datenpaket wird als Einzelsendung behandelt, und es gibt keine Rückmeldung darüber, ob ein Paket beim Empfänger angekommen ist.

Weil unter UDP aber keine Verbindungen auf- und abgebaut werden müssen und somit keine Timeout-Situationen entstehen können, kann UDP jedoch schneller als TCP sein: Wenn ein Paket verlorenght, wird die Datenübertragung hier eben ungehindert fortgesetzt, sofern nicht ein höheres Protokoll für Wiederholungen sorgt.

Die Datensicherheit ist unter UDP also in jedem Fall durch das Anwendungsprogramm zu gewährleisten.

Aufbau eines UDP-Datenpakets



- Source Port:** Portnummer der sendenden Anwendung (Rücksende-Port für Empfänger).
- Destination Port:** Zielport, an den die Daten beim Empfänger übertragen werden sollen.

Als Faustregel kann man sagen:

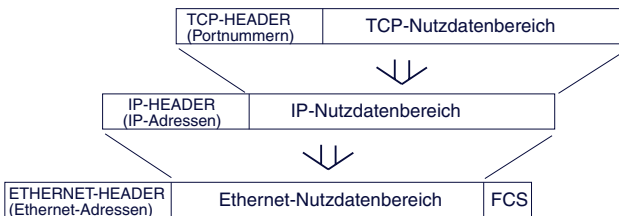
- Für kontinuierliche Datenströme oder große Datenmengen sowie in Situationen, in denen ein hohes Maß an Datensicherheit gefordert ist, wird in aller Regel TCP eingesetzt.
- Bei häufig wechselnden Übertragungspartnern sowie einer Gewährleistung der Datensicherheit durch übergeordnete Protokolle macht der Einsatz von UDP Sinn.

TCP/IP-Ethernet

TCP/IP ist ein rein logisches Protokoll und benötigt immer eine physikalische Grundlage. Wie bereits anfänglich erwähnt, genießt Ethernet heute die größte Verbreitung bei den physikalischen Netzwerktopologien. So findet man auch in den meisten TCP/IP-Netzwerken Ethernet als physikalische Grundlage.

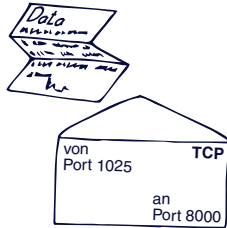
TCP/IP und Ethernet werden zusammengeführt, indem jedes TCP/IP-Paket in den Nutzdatenbereich eines Ethernet-Paketes eingebettet wird.

Aufbau eines TCP/IP-Ethernet-Datenpakets

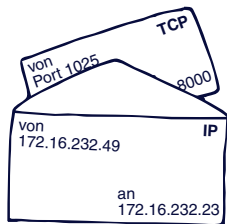


Die Nutzdaten passieren auf ihrem Weg von der Applikation auf dem PC bis ins Netzwerk mehrere Treiberebenen:

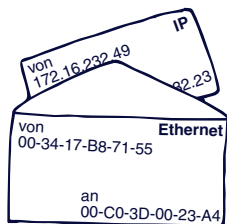
- Das Anwendungsprogramm entscheidet, an welchen anderen Netzteilnehmer die Daten gesendet werden sollen, und übergibt IP-Adresse und TCP-Port dem TCP/IP-Treiber (oft auch TCP/IP-Stack genannt).
- Der TCP/IP-Treiber koordiniert den Aufbau der TCP-Verbindung.
- Die vom Anwendungsprogramm übergebenen Nutzdaten werden vom TCP-Treiber je nach Größe in kleinere, übertragbare Blöcke geteilt.
- Jeder Datenblock wird zunächst vom TCP-Treiber in ein TCP-Paket verpackt.



- Der TCP-Treiber übergibt das TCP-Paket und die IP-Adresse des Empfängers an den IP-Treiber.
- Der IP-Treiber verpackt das TCP-Paket in ein IP-Paket.



- Der IP-Treiber sucht in der sogenannten ARP-Tabelle (Address Resolution Protocol) nach der Ethernet-Adresse des durch die IP-Adresse angegebenen Empfängers (dazu später mehr) und übergibt das IP-Paket zusammen mit der ermittelten Ethernet-Adresse an den Ethernet-Kartentreiber.
- Der Ethernet-Kartentreiber verpackt das IP-Paket in ein Ethernet-Paket und gibt dieses Paket über die Netzwerkkarte auf das Netzwerk aus.



Beim Empfänger findet die Prozedur in umgekehrter Reihenfolge statt:

- Die Ethernetkarte erkennt an der Destination-Ethernet-Adresse, daß das Paket für den Netzteilnehmer bestimmt ist und gibt es an den Ethernet-Treiber weiter.
- Der Ethernet-Treiber isoliert das IP-Paket und gibt es an den IP-Treiber weiter.
- Der IP-Treiber isoliert das TCP-Paket und gibt es an den TCP-Treiber weiter.
- Der TCP-Treiber überprüft den Inhalt des TCP-Paketes auf Richtigkeit und übergibt die Daten anhand der Portnummer an die richtige Applikation.

Auf den ersten Blick erscheint dieses vielschichtige Übertragungsverfahren ungeheuer umständlich. Aber erst die strikte Trennung von logischem Protokoll (TCP/IP) und physikalischem Protokoll (Ethernet), macht es möglich, netz-übergreifend und hardwareunabhängig Daten auszutauschen.

ARP – Address Resolution Protocol

Wie wir gesehen haben, übergibt der IP-Treiber neben dem IP-Datenpaket auch die physikalische Ethernet-Adresse an den Ethernetkarten-Treiber. Zur Ermittlung der Ethernet-Adresse des Empfängers bedient sich der IP-Treiber des Address Resolution Protocol ARP.

In jedem TCP/IP-fähigen Rechner gibt es eine ARP-Tabelle. Die ARP-Tabelle wird vom TCP/IP-Treiber bei Bedarf aktualisiert und enthält die Zuordnung von IP-Adressen zu Ethernet-Adressen.

| InternetAddress | Physical Address | Type |
|-----------------|-------------------|---------|
| 172.16.232.23 | 00-80-48-9c-ac-03 | dynamic |
| 172.16.232.49 | 00-c0-3d-00-26-a1 | dynamic |
| 172.16.232.92 | 00-80-48-9c-a3-62 | dynamic |
| 172.16.232.98 | 00-c0-3d-00-1b-26 | dynamic |
| 172.16.232.105 | 00-c0-3d-00-18-bb | dynamic |

Soll ein IP-Paket verschickt werden, sieht der IP-Treiber zunächst nach, ob die gewünschte IP-Adresse bereits in der ARP-Tabelle vorhanden ist. Ist dies der Fall, gibt der IP-Treiber die ermittelte Ethernet-Adresse zusammen mit seinem IP-Paket an den Ethernet-Kartentreiber weiter.

Kann die gewünschte IP-Adresse nicht gefunden werden, startet der IP-Treiber einen ARP-Request. Ein ARP-Request ist ein Rundruf (auch Broadcast genannt) an alle Teilnehmer im lokalen Netz.

Damit der Rundruf von allen Netzteilnehmern zur Kenntnis genommen wird, gibt der IP-Treiber als Ethernet-Adresse FF-FF-FF-FF-FF-FF an. Ein mit FF-FF-FF-FF-FF-FF adressiertes Ethernet-Paket wird grundsätzlich von allen Netzteilnehmern gelesen. Im IP-Paket wird als Destination die gewünschte IP-Adresse angegeben und im Feld Protocol des IP-Headers die Kennung für ARP ausgewiesen.

Derjenige Netzteilnehmer, der in diesem ARP-Request seine eigene IP-Adresse wiedererkennt, bestätigt das mit einem ARP-Reply. Der ARP-Reply ist ein sowohl auf Ethernet-, als auch

auf IP-Ebene an den ARP-Request-Absender adressiertes Datenpaket mit der ARP-Kennung im Protocol-Feld.

Der IP-Treiber kann nun die dem ARP-Reply entnommene Ethernet-Adresse der gewünschten IP-Adresse zuordnen und trägt sie in die ARP-Tabelle ein.

Im Normalfall bleiben die Einträge in der ARP-Tabelle nicht dauerhaft bestehen. Wird ein eingetragener Netzwerkteilnehmer über eine bestimmte Zeit (unter Windows ca. 2 Min.) nicht kontaktiert, wird der entsprechende Eintrag gelöscht. Das hält die ARP-Tabelle schlank und ermöglicht den Austausch von Hardwarekomponenten unter Beibehaltung der IP-Adresse. Man nennt diese zeitlich begrenzten Einträge auch dynamische Einträge.

Neben den dynamischen Einträgen gibt es auch statische Einträge, die der Benutzer selbst in der ARP-Tabelle ablegt. Die statischen Einträge können genutzt werden, um an neue Netzwerkkomponenten, die noch keine IP-Adresse haben, die gewünschte IP-Adresse zu übergeben.

Diese Art der Vergabe von IP-Adressen lassen auch Com-Server zu: Empfängt ein Com-Server, der noch keine eigene IP-Adresse hat, ein IP-Datenpaket, das auf Ethernet-Ebene an ihn adressiert ist, wird die IP-Adresse dieses Pakets ausgewertet und als eigene IP-Adresse übernommen.

Achtung: Nicht alle Netzwerkkomponenten besitzen diese Fähigkeit. PCs lassen sich auf diese Weise z.B. nicht konfigurieren!

Wir wissen nun, welche Informationen für eine TCP/IP-Ethernet-Verbindung im eigenen (lokalen) Netz benötigt werden. Was allerdings noch fehlt, sind diejenigen Informationen, die eine netzübergreifende Verbindung gestatten.

Gateway und Subnet-Mask

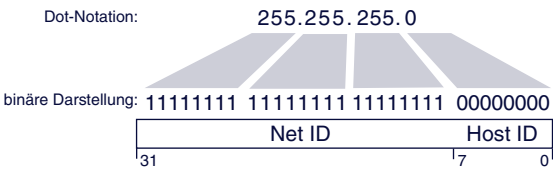
Ob der Empfänger, zu dem die Verbindung aufgebaut werden soll, im gleichen Netzwerk wie der Sender zu finden ist, erkennt man an der Net-ID – dem Teil der IP-Adresse, der das Netzwerk adressiert. Stimmt dieser Teil der IP-Adresse bei Sender und Empfänger überein, befinden sich beide im selben Netzwerk, stimmt er nicht überein, ist der Empfänger in einem anderen Netzwerk zu finden.

Die verschiedenen Einzelnetze werden über Gateways/Router miteinander verbunden und fügen sich so zum Internet zusammen.

Für die Netzwerkclassen A, B und C ist klar definiert, welcher Teil der IP-Adresse Net-ID und welcher Host-ID ist.

Nun ist es allerdings möglich, ein Netzwerk – egal welcher Netzwerkklasse –in weitere Unternetzwerke zu unterteilen. Zur Adressierung solcher Subnets reicht die von den einzelnen Netzwerkclassen vorgegebenen Net-ID allerdings nicht aus; man muss einen Teil der Host-ID zur Adressierung der Unternetze abzweigen. Im Klartext bedeutet dies, dass die Net-ID sich vergrößert und die Host-ID entsprechend kleiner wird.

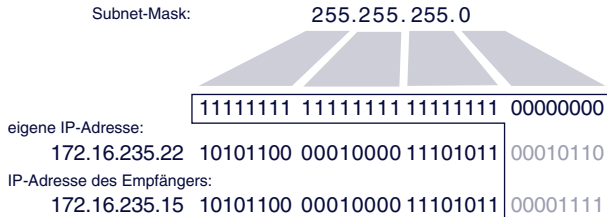
Welcher Teil der IP-Adresse als Net-ID und welcher als Host-ID ausgewertet wird, gibt die Subnet-Mask vor. Die Subnet-Mask ist genau wie die IP-Adresse ein 32-Bit-Wert, der in Dot-Notation dargestellt wird. Betrachtet man die Subnet-Mask in binärer Schreibweise, ist der Anteil der Net-ID mit Einsen, der Anteil der Host-ID mit Nullen aufgefüllt.



Bei jedem zu verschickenden Datenpaket vergleicht der IP-Treiber die eigene IP-Adresse mit der des Empfängers. Hierbei wer-

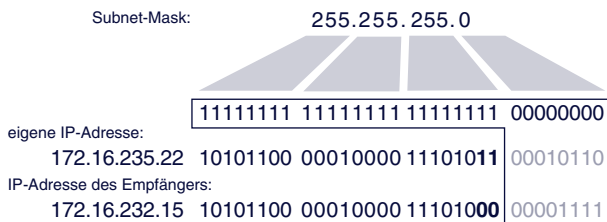
den die Bits der Host-ID über den mit Nullen aufgefüllten Teil der Subnet-Mask ausgeblendet.

Sind die ausgewerteten Bits beider IP-Adressen identisch, befindet sich der gewählte Netzteilnehmer im selben Subnet.



Im oben dargestellten Beispiel kann der IP-Treiber die Ethernet-Adresse über ARP ermitteln und diese dem Netzwerkkarten-Treiber zur direkten Adressierung übergeben.

Unterscheidet sich auch nur ein einziges der ausgewerteten Bits, befindet sich der gewählte Netzteilnehmer nicht im selben Subnet. In diesem Fall muss das IP-Paket zur weiteren Vermittlung ins Zielnetzwerk dem Gateway bzw. Router übergeben werden.



Im IP-Paket wird die IP-Adresse des gewünschten Netzteilnehmers eingetragen. Der IP-Treiber ermittelt über ARP aber nicht die Ethernet-Adresse des gewünschten Netzteilnehmers, sondern die Ethernet-Adresse des Routers.

Gateways bzw. Router sind im Prinzip nichts anderes als Computer mit zwei Netzwerkkarten. Ethernet-Datenpakete, die auf Karte A empfangen werden, werden vom Ethernet-Treiber

entpackt, und das enthaltene IP-Paket wird an den IP-Treiber weitergegeben. Dieser prüft, ob die Ziel-IP-Adresse zum an Karte B angeschlossenen Subnet gehört und das Paket direkt zugestellt werden kann, oder ob das IP-Paket an ein weiteres Gateway übergeben wird.

So kann ein Datenpaket auf seinem Weg von einem Netzteilnehmer zum anderen mehrere Gateways/Router passieren. Während auf IP-Ebene auf der gesamten Strecke die IP-Adresse des Empfängers eingetragen ist, wird auf Ethernet-Ebene immer nur das nächste Gateway adressiert. Erst auf dem Teilstück vom letzten Gateway/Router zum Empfänger wird in das Ethernet-Paket die Ethernet-Adresse des Empfängers eingesetzt.

Neben Routern, die ein Ethernet-Subnet mit einem anderen Ethernet-Subnet verbinden, gibt es auch Router, die das physikalische Medium wechseln –z.B. von Ethernet auf Token Ring oder ISDN. Während auch hier die IP-Adressierung über die gesamte Strecke gleich bleibt, ist die physikalische Adressierung von einem Router zum anderen, den auf den Teilstrecken geforderten physikalischen Gegebenheiten angepasst.

Zwischen zwei Ethernet-ISDN-Routern wird zum Beispiel über Telefonnummern adressiert.

Netzübergreifende TCP/IP-Verbindung

Im folgenden Abschnitt wird anhand einer bestehenden Telnetsverbindung der Weg eines Zeichens über eine geroutete Netzwerkverbindung beschrieben.

Wir gehen in unserem Beispiel davon aus, dass ein Anwender in Bremen bereits eine Telnetsverbindung zu einem W&T Com-Server in München aufgebaut hat; die Verbindung der Netze Bremen und München besteht in Form einer Routerverbindung über das ISDN-Netz.

Netzwerk Bremen

PC Bremen

| | |
|------------------|-------------------|
| IP-Adresse | 172.16.232.23 |
| Subnet-Mask | 255.255.255.0 |
| Gateway | 172.16.232.1 |
| Ethernet-Adresse | 03-D0-43-7A-26-A3 |



Router/Ethernet-Seite

| | |
|------------------|-------------------|
| IP-Adresse | 172.16.232.1 |
| Subnet-Mask | 255.255.255.0 |
| Gateway | |
| Ethernet-Adresse | 00-23-8B-47-99-01 |



Router/ISDN-Seite

| | |
|------------|--------------|
| Netzwerk | 172.16.232.0 |
| Telefonnr. | 0421 826217 |

Netzwerk München

Router/ISDN-Seite

| | |
|------------|--------------|
| Netzwerk | 190.107.43.0 |
| Telefonnr. | 089 99124711 |



Router/Ethernet-Seite

| | |
|------------------|-------------------|
| IP-Adresse | 190.107.43.1 |
| Subnet-Mask | 255.255.255.0 |
| Gateway | |
| Ethernet-Adresse | 00-23-8B-77-43-C0 |



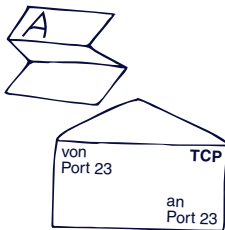
Com-Server München

| | |
|------------------|-------------------|
| IP-Adresse | 190.107.43.49 |
| Subnet-Mask | 255.255.255.0 |
| Gateway | 190.107.43.1 |
| Ethernet-Adresse | 00-0C-3D-00-32-04 |

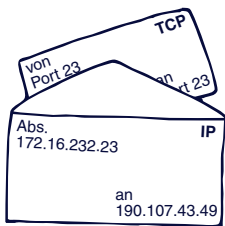
Der Anwender in Bremen gibt in der Telnets-Client-Anwendung das Zeichen „A“ ein.

- Das Telnets-Client-Programm auf dem PC übergibt dem TCP/IP-Stack das „A“ als Nutzdatum. Die IP-Adresse des Empfängers (190.107.43.49) und die Portnummer 23 für Telnets wurden dem TCP/IP-Stack bereits bei Aufbau der Verbindung übergeben.

- Der TCP-Treiber schreibt das „A“ in den Nutzdatenbereich eines TCP-Pakets und trägt als Destination-Port die 23 ein.



- Der TCP-Treiber übergibt das TCP-Paket und die IP-Adresse des Empfängers an den IP-Treiber.
- Der IP-Treiber verpackt das TCP-Paket in ein IP-Paket.



- Der IP-Treiber ermittelt über den Vergleich der Net-ID-Anteile von eigener IP-Adresse und IP-Adresse des Empfängers, ob das IP-Paket im eigenen Subnet zugestellt werden kann oder einem Router übergeben wird.

| | |
|----------------------------|---------------|
| Subnet-Mask: | 255.255.255.0 |
| eigene IP-Adresse: | 172.16.232.23 |
| IP-Adresse des Empfängers: | 190.107.43.49 |

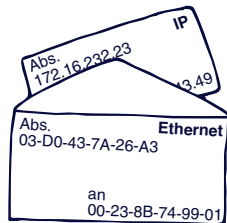
| | | | |
|----------|----------|----------|----------|
| 11111111 | 11111111 | 11111111 | 00000000 |
| 10101100 | 00010000 | 11101000 | 00010111 |
| 01101011 | 00101011 | 00110001 | |

Hier sind die Net-ID-Anteile der beiden Adressen nicht gleich; das IP-Paket muss folglich an den eingetragenen Router übergeben werden.

- Der IP-Treiber ermittelt über ARP die Ethernet-Adresse des Routers. Da die TCP-Verbindung bereits aufgebaut ist, wird die IP-Adresse des Routers bereits in der ARP-Tabelle aufgelöst sein.

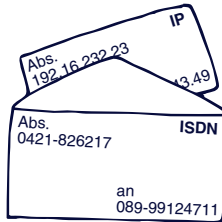
| Internet Address | Physical Address | Type |
|------------------|-------------------|---------|
| → 172.16.232.1 | 00-23-8B-74-99-01 | dynamic |
| 172.16.232.49 | 00-c0-3d-00-26-a1 | dynamic |
| 172.16.232.92 | 00-80-48-9c-a3-62 | dynamic |

- Der IP-Treiber entnimmt der ARP-Tabelle die Ethernet-Adresse des Routers und übergibt sie zusammen mit dem IP-Paket dem Ethernet-Kartentreiber.
- Der Ethernet-Kartentreiber verpackt das IP-Paket in ein Ethernet-Paket und gibt dieses Paket über die Netzwerkkarte auf das Netzwerk aus.

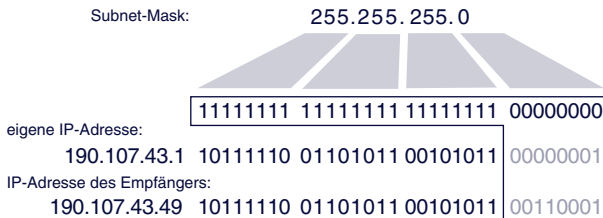


- Der Router entnimmt dem empfangenen Ethernet-Paket das IP-Paket.
- Die IP-Adresse des Empfängers wird mit einer sogenannten Routing-Tabelle verglichen. Anhand dieser Routing-Tabelle entscheidet der ISDN-Router, über welche Rufnummer das gesuchte Netzwerk zu finden ist. Da die TCP-Verbindung bereits besteht, ist vermutlich auch die ISDN-Verbindung zu diesem Zeitpunkt schon aufgebaut. Sollte dies nicht mehr der Fall sein, wählt der Router die der Routing-Tabelle entnommene Rufnummer und stellt die ISDN-Verbindung zum Gegen-Router im Zielnetzwerk wieder her.

- Auch im ISDN-Netz wird das IP-Paket in einen Rahmen von Adressinformationen eingepackt. Für uns ist nur wichtig, dass es in seinem Adressierungsbereich unverändert in das ISDN-Paket übernommen wird.

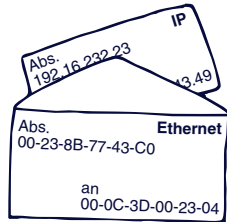


- Der Router im Zielnetz entnimmt dem empfangenen ISDN-Paket das IP-Paket. Über IP-Adressen und Subnet-Mask wird festgestellt, ob das empfangene IP-Paket im lokalen Subnet zugestellt werden kann oder einem weiteren Router übergeben werden muss.



In unserem Beispiel hat das IP-Paket das Zielnetzwerk erreicht und kann im lokalen Netz über Ethernet adressiert werden.

- Der Router, der intern ebenfalls eine ARP-Tabelle führt, ermittelt über ARP die zur IP-Adresse passende Ethernet-Adresse und verpackt das im Adressierungsbereich immer noch unveränderte IP-Paket in ein Ethernet-Paket.



- Der Com-Server erkennt an der Destination-Ethernet-Adresse, dass das Paket für ihn bestimmt ist, und entnimmt das IP-Paket.
- Der IP-Treiber des Com-Servers isoliert das TCP-Paket und gibt es an den TCP-Treiber weiter.
- Der TCP-Treiber überprüft den Inhalt des TCP-Paketes auf Richtigkeit und übergibt die Daten – in diesem Fall das „A“ – an den seriellen Treiber.
- Der serielle Treiber gibt das „A“ auf der seriellen Schnittstelle aus.

Bei einer TCP-Verbindung wird der korrekte Empfang eines Datenpaketes mit dem Rücksenden einer Acknowledgement-Nummer quittiert. Das Quittungspaket durchläuft den gesamten Übertragungsweg und alle damit verbundenen Prozeduren in Gegenrichtung. All dies spielt sich innerhalb weniger Millisekunden ab.

DHCP – Dynamic Host Configuration Protocol

Zur Erinnerung: Jedes Ethernet-Endgerät hat eine weltweit einmalige Ethernet-Adresse (MAC-Adresse), die vom Hersteller vorgegeben und nicht veränderbar ist. Für den Einsatz in TCP/IP-Netzen vergibt der Netzwerkadministrator dem Endgerät zusätzlich eine zum Netzwerk passende IP-Adresse.

Wird kein DHCP benutzt, werden die IP-Adressen „klassisch“ vergeben:

- Bei Geräten, die direkte User-Eingaben erlauben (z.B. PCs), kann die IP-Nummer direkt in ein entsprechendes Konfigurationsmenü eingegeben werden.
- Bei „Black-Box-Geräten“ (z.B. Com-Servern) gibt es zum einen das ARP-Verfahren über das Netzwerk, zum anderen besteht die Möglichkeit, die Konfigurationsinformation über eine serielle Schnittstelle einzugeben.

Neben der IP-Adresse müssen als weitere Parameter noch Subnet-Mask und Gateway sowie ggf. ein DNS-Server (mehr dazu im nächsten Kapitel) konfiguriert werden. Bei großen Netzen mit vielen unterschiedlichen Endgeräten bringt das allerdings schnell ein hohes Maß an Konfigurations- und Verwaltungsaufwand mit sich.

Mit DHCP wird dem Netzwerkadministrator ein Werkzeug angeboten, mit dem die Netzwerkeinstellungen der einzelnen Endgeräte automatisch, einheitlich und zentral konfigurierbar sind.

Für die Nutzung von DHCP wird im Netzwerk mindestens ein DHCP-Server benötigt, der die Konfigurationsdaten für einen vorgegebenen IP-Adressbereich verwaltet. DHCP-fähige Endgeräte erfragen beim Booten von diesem Server ihre IP-Adresse und die zugehörigen Parameter wie Subnet-Mask und Gateway. DHCP-Server sehen drei grundsätzliche Möglichkeiten der IP-Adresszuteilung und Konfiguration vor:

Vergabe der IP-Adresse aus einem Adresspool

Auf dem DHCP-Server wird ein Bereich von IP-Adressen festgelegt, aus dem einem anfragenden Netzteilnehmer eine zur Zeit nicht benutzte Adresse zugeteilt wird. Die Zuteilung ist bei diesem Verfahren in aller Regel zeitlich begrenzt, wobei die Nutzungsdauer (Lease-Time) vom Netzwerkadministrator festgelegt oder ganz deaktiviert werden kann. Darüber hinaus lassen sich wichtige Daten (Lease-Time, Subnet-Mask, Gateway, DNS-Server usw.) in einem Konfigurationsprofil hinterlegen, das für alle Endgeräte gilt, die aus dem Adresspool bedient werden.

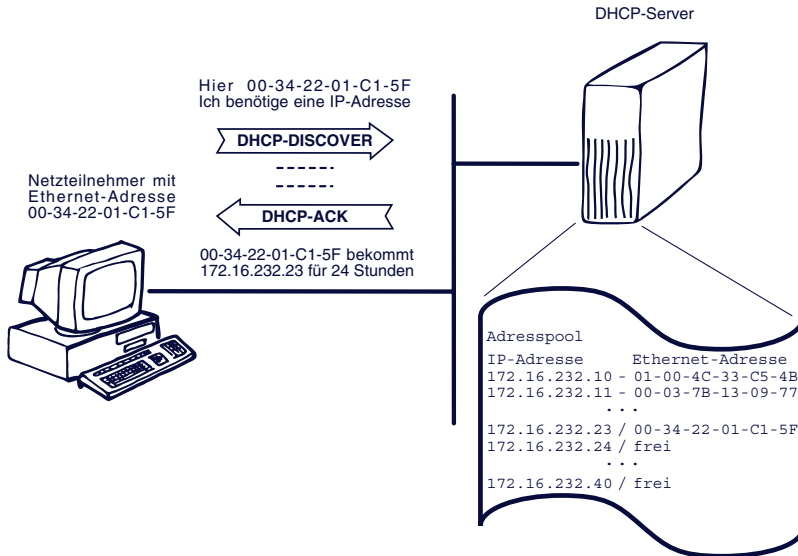
Vorteile Geringer Administrationsaufwand; Anwender können mit dem selben Endgerät ohne Konfigurationsaufwand an verschiedenen Standorten ins Netzwerk.

Sofern nicht alle Endgeräte gleichzeitig im Netzwerk aktiv sind, kann die Anzahl der möglichen Endgeräte größer sein als die Zahl der verfügbaren IP-Adressen.

Nachteile Ein Netzteilnehmer kann nicht an Hand seiner IP-Adresse identifiziert werden, da die eindeutige Zuordnung von IP-Adresse und Endgerät verloren geht.

Es kann vorkommen, dass ein Endgerät bei jedem Start eine andere IP-Adresse zugewiesen bekommt.

Beispiel: Typische Fälle für die Vergabe von IP-Adressen aus einem Adresspool sind Universitätsnetzwerke. Hier gibt es Netze mit einer fast unbegrenzten Zahl potentieller Anwender, von denen aber nur jeweils wenige tatsächlich im Netzwerk arbeiten. Dank DHCP haben die Studenten die Möglichkeit, ihr Notebook ohne Konfigurationsänderung von einem Labor ins andere mitzunehmen und im Netzwerk zu betreiben.



Vergabe einer reservierten IP-Adresse

Der Netzwerkadministrator hat die Möglichkeit, einzelne IP-Adressen für bestimmte Endgeräte zu reservieren. Auf dem DHCP-Server wird dazu der IP-Adresse die Ethernet-Adresse des Endgeräts zugeordnet; für jede reservierte IP-Adresse kann außerdem ein individuelles Konfigurationsprofil angelegt werden. Die Angabe einer Lease-Time ist in diesem Fall nicht sinnvoll (aber trotzdem möglich), da die IP-Adresse ohnehin nur vom zugeordneten Endgerät benutzt wird.

Vorteile: Trotz individueller Konfiguration lassen sich alle Netzwerkeinstellungen an zentraler Stelle erledigen und müssen nicht am Endgerät selbst vorgenommen werden.

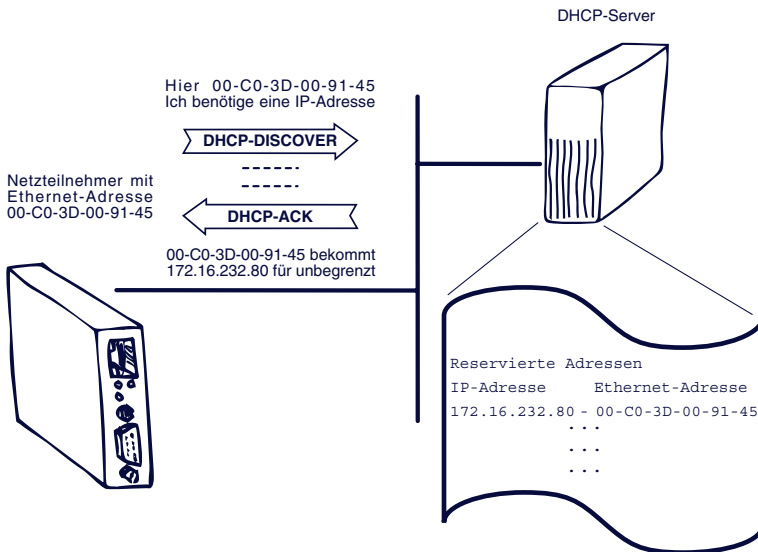
Endgeräte können gezielt über Ihre IP-Adresse angesprochen werden.

Nachteile: Da für jedes Endgerät spezifische Einstellungen angegeben werden müssen, steigt der Administrationsaufwand.

Beim Austausch von Endgeräten muss auf dem DHCP-Server im Konfigurationsprofil mindestens die Ethernet-Adresse neu eingetragen werden.

Beispiel: Konfiguration von DHCP-fähigen Endgeräten wie Printservern oder Com-Servern, bei denen je nach Einsatzfall eine Adressierung über IP-Adresse benötigt wird. Im DHCP-Manager wird bei der reservierten IP-Adresse die Ethernet-Adresse des zugehörigen Endgerätes eingetragen; die Lease-Time sollte deaktiviert sein. Beim Com-Server können als zusätzliche Parameter Subnet-Mask und Gateway (Router) angegeben werden.

Hierzu muss ergänzend gesagt werden, dass viele Endgeräte auch das ältere BootP-Protokoll nutzen, um ihre Konfiguration zu erfragen. BootP ist ein Vorläufer von DHCP und wird von DHCP-Servern unterstützt. Allerdings kann BootP nur mit reservierten IP-Adressen arbeiten.



Bei „Black-Box-Geräten“ wie dem Com-Server sollte das BootP-Protokoll eingesetzt werden, um in jedem Fall die Übergabe einer reservierten IP-Adresse zu erzwingen. Ist beim DHCP-Server kein zur Ethernet-Adresse des Com-Server passender Eintrag vorhanden, wird die BootP-Anfrage ignoriert und der Com-Server behält die aktuell eingestellte IP-Adresse.

Ausschluss bestimmter IP-Adressen aus der DHCP-Konfiguration

Für Endgeräte, die weder DHCP- noch BootP-fähig sind, hat der Netzwerkadministrator die Möglichkeit, einzelne IP-Adressen oder auch ganze Adressbereiche von der Vergabe durch DHCP auszuschließen.

Die Konfiguration muss in diesem Fall entweder am Endgerät selbst vorgenommen werden oder durch den Einsatz mitgelieferter Tools erfolgen.

Nachteil: Uneinheitliche und ggf. dezentrale Konfiguration; ein Höchstmaß an Administrationsaufwand ist erforderlich.

Beispiel: PCs mit älteren DOS-Versionen oder ältere Printserver und Com-Server sind nicht DHCP-fähig und müssen auf jeden Fall „von Hand“ konfiguriert werden.

Alle drei Verfahren können in Netzwerken mit DHCP-Unterstützung nebeneinander angewandt werden.

DHCP und Router

Der Informationsaustausch zwischen Endgeräten und DHCP-Servern erfolgt auf physikalischer Ebene in Form von UDP-Broadcasts (Rundrufen ins Netz). Erstreckt sich die DHCP-Konfiguration über mehrere Subnetze, sollte bei der Auswahl geeigneter Router darauf geachtet werden, dass diese auch DHCP-Broadcasts weiterleiten.



DHCP-Server unter Windows 2000 vergeben auch auf BootP-Anfragen hin IP-Adressen aus dem normalen Adress-Pool. Diese Eigenschaft lässt sich aber deaktivieren, was Ihr Netzwerk-administrator unbedingt tun sollte!

DNS – das Domain Name System

Das Domain Name System ist das Adressbuch des Internet. Obwohl es vom Anwender nur im Hintergrund genutzt wird, ist es doch einer der wichtigsten Internetdienste.

Auf IP-Ebene werden die Millionen von Teilnehmern im Internet über IP-Adressen angesprochen. Für den Nutzer wäre der Umgang mit IP-Adressen aber schwierig: Wer kann sich schon merken, dass das Web-Thermometer von W&T unter der IP-Adresse 195.8.247.225 zu erreichen ist? Einen aussagekräftigen Namen, wie *www.klima.wut.de*, kann man sich dagegen viel leichter merken.

Schon in den Anfängen des Internet trug man dem Bedürfnis Rechnung, IP-Adressen symbolische Namen zuzuordnen: auf jedem lokalen Rechner wurde eine *Hosts*-Tabelle gepflegt, in der die entsprechenden Zuordnungen hinterlegt waren. Der Nachteil bestand jedoch darin, dass eben nur diejenigen Netzwerkteilnehmer erreichbar waren, deren Namen in der lokalen Liste standen. Zudem nahmen diese lokalen Listen mit dem rapiden Wachstum des Internet bald eine nicht mehr handhabbare Größe an. Man stand also vor der Notwendigkeit, ein einheitliches System zur Namensauflösung zu schaffen. Aus diesem Grund wurde 1984 der DNS-Standard verabschiedet, an dem sich bis heute kaum etwas geändert hat.

Das Prinzip ist einfach. Die Zuordnung von IP-Adressen und Domainnamen wird auf sogenannten DNS-Servern hinterlegt und dort bei Bedarf „angefragt“. Doch ehe wir hier in die Details gehen, noch einige Anmerkungen zum Aufbau von Domainnamen:

Domainnamen

Das DNS sieht eine einheitliche Namensvergabe vor, bei der jeder einzelne Host (Teilnehmer im Netz), Teil mindestens einer übergeordneten „Top-Level-Domain“ ist.

Als Top-Level-Domain bietet sich ein länderspezifischer Domainname an:

- *.de* für Deutschland
- *.at* für Österreich
- *.ch* für Schweiz usw.

Die Domain kann aber auch nach Inhalt bzw. Betreiber gewählt werden:

- *.com* für kommerzielle Angebote
- *.net* für Netzbetreiber
- *.edu* für Bildungseinrichtungen
- *.gov* ist der US-Regierung vorbehalten
- *.mil* ist dem US-Militär vorbehalten
- *.org* für Organisationen

Alle untergeordneten (Sub-Level-) Domainnamen können vom Betreiber selbst gewählt werden, müssen in der übergeordneten Domain aber einmalig sein. Für jede Top-Level-Domain gibt es eine selbstverwaltende Institution, bei der die Sub-Level-Domains beantragt werden müssen und die damit eine Mehrfachvergabe ausschließt. Für die *de*-Domain ist in solchen Fragen die DENIC (*Deutsches Network Information Center*; <http://www.denic.de>) zuständig.

Ein Beispiel: *www.klima.wut.de* setzt sich zusammen aus:

- *de* für Deutschland als Top-Level-Domain
- *wut* für Wiesemann und Theis als Sub-Level-Domain
- *www.klima* für das Web-Thermometer in der Domain *wut.de*

Der gesamte Domainname darf maximal 255 Zeichen lang sein, wobei jeder Subdomainname höchstens 63 Zeichen umfassen darf. Die einzelnen Subdomainnamen werden mit Punkten getrennt. Eine Unterscheidung zwischen Groß- und Kleinschreibung gibt es nicht. *WWW.WUT.DE* führt Sie genauso auf die Homepage von W&T wie *www.wut.de* oder *www.WuT.de*.

Namensauflösung im DNS

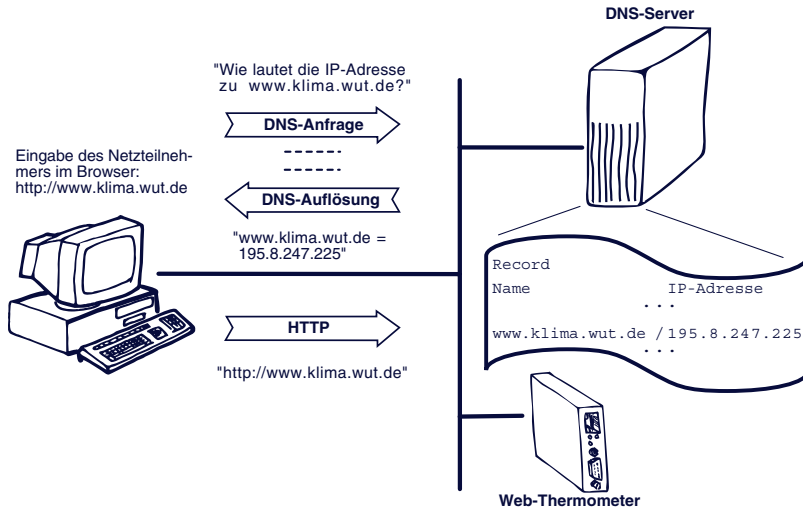
Wie bereits angesprochen, werden auf DNS-Servern (auch Nameserver genannt) Listen mit der Zuordnung von Domainnamen und IP-Adresse geführt. Gäbe es bei den heutigen Ausdehnungen des Internet nur einen einzigen DNS-Server, wäre dieser vermutlich mit der immensen Zahl der DNS-Anfragen hoffnungslos überfordert. Aus diesem Grund wird das Internet in Zonen aufgeteilt, für die ein bzw. mehrere DNS-Server zuständig sind.

Netzteilnehmer, die das DNS nutzen möchten, müssen in ihrem TCP/IP-Stack die IP-Adresse eines in Ihrer Zone liegenden DNS-Servers angeben. Um auch bei Ausfall dieses Servers arbeiten zu können, verlangen die üblichen TCP/IP-Stacks sogar die Angabe eines zweiten DNS-Servers.

Welcher DNS-Server für den jeweiligen Netzteilnehmer zuständig ist, erfährt man beim Provider bzw. beim Netzwerk-administrator.

Um Domainnamen in IP-Adressen auflösen zu können, verfügen heutige TCP/IP-Stacks über ein Resolver-Programm. Gibt der Anwender anstatt einer IP-Adresse einen Domainnamen an, startet das Resolver-Programm eine Anfrage beim eingetragenen DNS-Server. Liegt dort kein Eintrag für den gesuchten Domainnamen vor, wird die Anfrage an den in der Hierarchie nächsthöheren DNS-Server weitergegeben. Dies geschieht so lange, bis die Anfrage entweder aufgelöst ist oder festgestellt wird, dass es den angefragten Domainnamen nicht gibt.

Die zum Domainnamen gehörende IP-Adresse wird von DNS-Server zu DNS-Server zurückgereicht und schließlich wieder dem Resolver-Programm übergeben. Der TCP/IP-Stack kann nun die Adressierung des Zielteilnehmers ganz normal über dessen IP-Adresse vornehmen.



Die Zuordnung von IP-Adresse und Domainnamen wird vom TCP/IP-Stack in einem Cache hinterlegt. Diese Cache-Einträge sind dynamisch: Wird der hinterlegte Netzteilnehmer für bestimmte Zeit nicht angesprochen, löscht der Stack den Eintrag wieder. Das hält den Cache schlank und macht es möglich, die zu einem Domainnamen gehörende IP-Adresse bei Bedarf auszutauschen.

DNS in Embedded-Systemen

Embedded-Systeme bieten in aller Regel nicht die Möglichkeit, am Gerät selbst einen Domainnamen einzugeben.

Das ist auch gar nicht nötig, denn das Endgerät muss seinen eigenen Namen gar nicht wissen. Vielmehr wird die Zuordnung von Name und IP-Adresse auch hier auf dem DNS-Server festgehalten. Soll z.B. von einem Client eine Verbindung auf ein als Server arbeitendes Embedded-System aufgebaut werden, erfragt der Client die zum Namen gehörende IP-Adresse wie gehabt beim DNS-Server.

Da Embedded-Systeme aber häufiger in „Maschine-Maschine-Verbindungen“ als in „Mensch-Maschine-Verbindungen“ ar-

beiten, kann eine direkte Adressierung über IP-Adresse hier effizienter sein, da die Zeit für die DNS-Auflösung entfällt.

Die Adressierung über Namen ist bei Embedded-Systemen nur dann sinnvoll, wenn entweder nur der Name bekannt ist (z.B. E-Mail-Adressen) oder mit einem „Umzug“ eines Servers (Name bleibt, IP-Adresse ändert sich) gerechnet werden muss (z.B. Webserver).

DHCP und DNS

Während DHCP die Zuordnung von physikalischem Endgerät, also der Ethernet-Adresse zur IP-Adresse verwaltet, sind auf DNS-Servern die zu den Domainnamen gehörenden IP-Adressen hinterlegt. Obwohl gerade bei der dynamischen IP-Adressvergabe via DHCP ein automatischer Abgleich mit DNS Sinn machen würde, findet dieser leider nicht statt.

Für fest zugeteilte IP-Adressen können die Eintragungen auf DHCP- und DNS-Servern von Hand oder durch zusätzliche Softwaretools synchronisiert werden. Bei dynamisch zugewiesenen IP-Adressen kann die Pflege der zugehörigen DNS-Einträge nur durch den Einsatz zusätzlicher Tools realisiert werden.

Für UNIX-Systeme gibt es DHCP-Server, die im Vorfeld der eigentlichen Adressvergabe eine Zuordnung von Ethernet-Adressen zu Namen statt Ethernet-Adressen zu IP-Nummern erlauben. Dem DNS-Server bleibt so die Zuordnung von Name und IP-Adresse vorbehalten. Die Adressvergabe verläuft dann folgendermaßen:

1. Das Endgerät versucht, vom DHCP-Server eine IP-Adresse zu beziehen.
2. Anhand der Ethernet-Adresse des Endgeräts findet der DHCP-Server den zum Endgerät gehörenden Namen.
3. Der DHCP-Server erfragt beim DNS-Server die zum Namen gehörende IP-Adresse.
4. Der DHCP-Server übergibt dem Endgerät die vom DNS-Server aufgelöste IP-Adresse.

Für die nächste Generation von Betriebssystemen ist die Einführung eines erweiterten DNS, dem dynamischen DNS (kurz DDNS) geplant. Mit DDNS soll es einen Abgleich zwischen DNS und DHCP geben, so dass auch Endgeräte, die ihre Adresse aus einem Adresspool beziehen, über Domainnamen ansprechbar sind.

Leider gab es bei Drucklegung nur Ankündigungen, aber noch keine fundierten Informationen zu diesem Thema.

Weitere Protokolle und Dienste

Die grundlegende Funktionalität von TCP/IP-Ethernet wäre nun geklärt. Doch in Netzwerken kann Ihnen noch eine ganze Reihe weiterer Protokolle und Dienste begegnen.

In diesem Abschnitt erfahren Sie, wie E-Mail funktioniert, was beim Aufruf einer Webseite geschieht, und welche wichtigen Protokolle und Dienste Ihnen im Zusammenhang mit TCP/IP-Ethernet sonst noch begegnen können.

WWW – World Wide Web

In den ersten 20 Jahren seines Daseins war die Nutzung des Internets für normale Menschen kaum interessant. Eine für heutige Verhältnisse kleine Gruppe von Insidern musste kryptische Befehlszeilen eintippen, um Informationen auszutauschen zu können.

Erst mit der Entwicklung des WWW-Standards erschloss sich das Internet einem immer breiter werdenden Publikum. Um die Möglichkeiten des WWW nutzen zu können, benötigt der Anwender einen Internetbrowser – ein Client-Programm, das über eine graphische Oberfläche die Inhalte von Webseiten anzeigt, die auf WWW-Servern hinterlegt sind.

Übersetzt man das englische Wort „browse“ ins Deutsche, heißt es soviel wie „Schmökern, Blättern“. Und genau das ist die Philosophie, die sich hinter WWW verbirgt.

Der Anwender soll bei minimalen Bedienungsaufwand durch ein riesiges Netz (eng. Web) von Informationen navigieren können. Und das alles ganz einfach per Mausklick: ausgeprägte Computer und Netzwerkkenntnisse sind dazu nicht notwendig.

Webseiten liegen als Hypertext vor und können neben Textinformationen auch Verweise auf Bilder, Graphiken und weitere multimediale Inhalte enthalten. Wie all diese Elemente im Browser angezeigt werden sollen, ist ebenfalls im Hypertext hinterlegt.

Die wichtigste Errungenschaft im WWW ist aber die „Verlinkung“ von Inhalten. Jedes Element einer Webseite kann mit einem Hyperlink – einem Verweis auf eine andere Webseite – versehen werden. Klickt der Anwender mit der Maus ein solches verlinktes Element an, öffnet sich im Browser automatisch die ausgewählte Webseite. Der Anwender kann also per Mausklick in einem Netz von Seiten und anderen Inhalten hin und her springen (browsen).

Die wesentliche Grundlage für das World Wide Web bilden drei Dinge:

- **URL – Uniform Resource Locator**
Über den URL gibt der Anwender dem Browser vor, welches Protokoll genutzt wird, auf welchem Webserver die Seite liegt, und wo diese auf dem Webserver zu finden ist
- **HTML – Hypertext Markup Language**
Eine Seitenbeschreibungssprache, die über Schlüsselwörter vorgibt, wie die Inhalte im Browser angezeigt werden, wo Multimedia-Elemente zu finden sind und welche Elemente wie verlinkt sind.
- **HTTP – Hypertext Transfer Protocol**
Das HTTP-Protokoll regelt die Anforderung und Übertragung von Webinhalten zwischen HTTP-Server und Browser.

URL - Uniform Resource Locator

Eine Voraussetzung dafür, dass der Nutzer sich im WWW gut zurechtfindet, ist ein einheitliches Adressierungsschema. Diese Aufgabe übernimmt der URL, der generell folgendes Format hat:

```
protokoll://hostname [:tcp-port] [/pfadname] [/filename] [?weitere parameter]
```

Die Angabe von Protokoll und Hostname ist in jedem Fall erforderlich; alle weiteren Parameter sind optional.

Protokoll

Von den, durch die meisten Browser unterstützten Protokollen, gehen wir hier nur auf die drei wichtigsten ein:

HTTP wird für den Zugriff auf Webseiten benutzt und ist somit *das* WWW-Protokoll schlechthin.

Beispiel: *http://www.wut.de* öffnet die Homepage von Wiesemann & Theis

FTP dient zur Dateiübertragung und wird zum Up- und Download ganzer Dateien verwendet.
Beispiel *ftp://www.wut.de/download/anleitg/tcpip_anf.pdf* startet den Download von TCPIP_ANF.PDF. Bei FTP ist zu berücksichtigen, dass der Nutzer ggf. bestimmte Zugriffsrechte benötigt, um die gewünschte Aktion auszuführen.

Telnet veranlasst den Browser, einen Telnet-Client zu öffnen und eine Telnetverbindung zum angegebenen Host aufzubauen. Telnet wird häufig genutzt, um Embedded-Systeme via Netzwerk zu konfigurieren. Gibt man im Browser als URL z.B. *telnet://<IP-Adresse eines W&T Com-Servers>:1111* an, wird man sofort mit dem Konfigurationsport des Geräts verbunden.

Hostname

Hier wird der Hostname oder die IP-Adresse des Servers angegeben, mit dem die Verbindung hergestellt werden soll. Der Hostname des W&T Webserver lautet beispielsweise *http://www.wut.de*.

Bei Protokoll und Hostname spielt es keine Rolle, ob groß oder klein geschrieben wird.

HTTP://www.wut.de führt genauso auf die Homepage von W&T wie *http://www.WuT.de*.

TCP-Port

Einige Standardprotokolle unter TCP greifen normalerweise auf fest zugeordnete TCP-Ports zu:

| | |
|---------------|---------|
| HTTP | Port 80 |
| FTP | Port 21 |
| Telnet | Port 23 |

W&T

Möchte der Anwender die Verbindung auf einen anderen Port aufbauen, kann er das im URL über den Parameter TCP-Port tun.

Beispiel: Mit *telnet://<IP-Adresse eines W&T Com-Servers>:1111* wird der Zugriff auf den Konfigurationsport eines W&T Com-Servers eingeleitet.

Pfadname

Auf einem WWW-Server können die Inhalte genau wie auf einem lokalen Rechner in verschiedenen Unterverzeichnissen abgelegt werden. Der Pfadname gibt also an, wo auf dem Server der gewünschte Inhalt zu finden ist.

Filename

Gibt den Namen der Datei an, auf die zugegriffen werden soll.

Bei Pfadname und Filename muß zwischen Groß- und Kleinschreibung unterschieden werden!

Verzichtet der Anwender bei Verwendung des HTTP-Protokolls darauf, einen Dateinamen anzugeben, wird automatisch auf eine Datei namens *index.html* oder *default.html* zugegriffen, sofern diese vorhanden ist. Beispiel: *http://www.wut.de* entspricht *http://www.wut.de/index.html*.

Weitere Parameter

Alle Angaben nach einem Fragezeichen werden als Parameter an eine auf dem WWW-Server laufende Applikation übergeben (hierzu später mehr).

HTML – Hypertext Markup Language

Eines der Probleme im WWW war zunächst die Vielzahl unterschiedlicher Rechner und Betriebssysteme. Eine einheitliche Softwareschnittstelle auf Anwenderebene gab es nicht. Aus dem Bedürfnis heraus, eine auch für den Laien einfach zu bedienende Oberfläche zu schaffen, die sich auf verschiedensten Rechnern gleich darstellt, wurde HTML entwickelt.

HTML ist eine Auszeichnungssprache (Markup Language) die sich aus Schlüsselwörtern – auch Tags genannt – und den darzustellenden Inhalten zusammensetzt. Die Tags geben an, in welcher Art und Weise der nachfolgende Text darzustellen ist. So lassen sich z.B. Schriftgröße, -art und -ausrichtung vorgeben, Inhalte können in Tabellen oder in Form einer numerischen Aufzählung dargestellt werden, die Farbe von Text und Hintergrund kann festgelegt werden usw.

Neben Text können mit Hilfe von HTML auch Grafiken angezeigt werden, und sogar multimediale Inhalte wie Musik, Sprache oder Filmsequenzen lassen sich per HTML einbinden. Das HTML-Dokument selbst transportiert dabei ausschließlich Textinhalte. Für jedes andere darzustellende Element wird via HTML angegeben, von wo es geladen werden kann, wo es auf dem Bildschirm erscheinen soll und in welcher Größe es dargestellt werden soll.

Die wohl wichtigste Eigenschaft von HTML liegt darin, dass alle Elemente mit einem Verweis – auch *Hyperlink* oder kurz *Link* genannt – versehen werden können. Klickt der Anwender ein solches Element mit der Maus an, wird er automatisch auf eine weitere Website weiter geleitet, bekommt eine Grafik angezeigt oder startet einen Download.

Mit der Erklärung der von HTML bereitgestellten Tags lassen sich ganze Bücher füllen. Deshalb beschränken wir uns hier auf die elementaren Tags und Eigenschaften von HTML.

Für HTML-Tags gilt ein festes Schema:

- Einzelne Tags sind in spitze Klammern „eingepackt“. *<HTML-Tag>*
- Das eigentliche Tag kann durch Angabe von Attributen erweitert werden. *<HTML-Tag Attribut="xy">*
- Zu jedem HTML-Tag gibt es ein entsprechendes Ende-Tag, das durch einen voran gestellten Schrägstrich (Slash) gekennzeichnet ist. *</HTML-Tag>*
- Die durch ein Tag definierten Eigenschaften gelten für alles, was zwischen dem Tag und dem Ende-Tag steht. *<HTML-Tag> Gültigkeitsbereich </HTML-Tag>*
- Bei HTML-Tags wird nicht zwischen Groß- und Kleinschreibung unterschieden. *<HTML>* ist gleichbedeutend mit *<html>*.

Grundsätzlicher Aufbau einer HTML-Datei

Jede HTML-Datei wird mit *<HTML>* eingeleitet und endet mit *</HTML>*. Man unterscheidet beim weiteren Aufbau einer Seite zwischen Kopf und Körper.

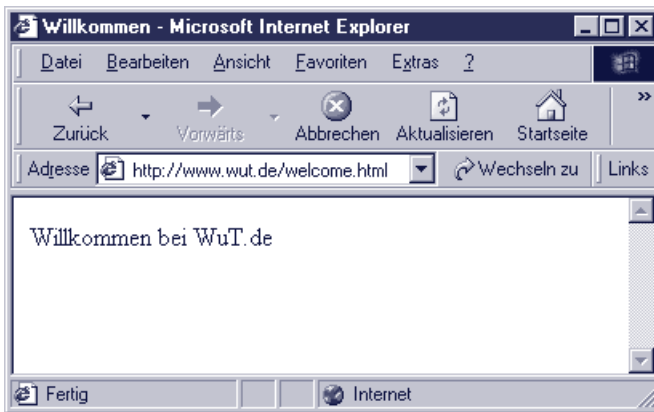
Alle Angaben im Kopf bleiben für den Betrachter unsichtbar und enthalten Eigenschaften der Seite, die nicht direkt die Darstellung betreffen. Einzige Ausnahme ist der Titel, der in der Titelleiste des Browserfensters angezeigt wird. Die Kopfinformationen stehen zwischen den Tags *<head>* und *</head>*

Auf den Kopf folgt der Seitenkörper, der mit dem *<body>*-Tag eingeleitet wird. Im Körper der HTML-Seite sind alle Angaben zu finden, die den eigentlichen Inhalt der Seite und dessen Darstellung betreffen. Das Ende des Körpers wird mit dem *</body>* Tag gekennzeichnet.

Hier ein einfaches Beispiel:

```
<html>
  <head>
    <title>Willkommen</title>
  </head>
  <body bgcolor="#FFFFFF">
    Willkommen bei WuT.de
  </body>
</html>
```

Bitte beachten Sie, dass beim `<body>`-Tag das Attribut `bgcolor="#FFFFFF"` für einen weißen Hintergrund angegeben wurde. Im Browser sieht das dann so aus:



Hyperlinks

Einer der großen Vorteile von HTML liegt in der Möglichkeit, einzelne Inhaltselemente mit einem einen Hyperlink zu versehen. Klickt der Anwender auf ein solches verlinktes Element, wird er auf eine andere Webseite weitergeleitet.

Wir erweitern unseren HTML-Code um einen Hyperlink:


```
<body bgcolor="#FFFFFF">  
    Willkommen bei <a href="http://www.wut.de/index.html">WuT.de</a>  
</body>
```

Bei einem Mausklick auf „WuT.de“ werden wir nun auf die Homepage von W&T gelenkt.

Das Pfadattribut des Tags `` kann die Pfadangabe entweder in absoluter oder in relativer Form enthalten.

- Absolut: es wird der komplette URL angegeben, auf den der Hyperlink verweisen soll.
- Relativ: es wird nur der Name der Datei angegeben auf die zugegriffen werden soll. Die Datei wird dann im gleichen Verzeichnis gesucht, in dem sich auch die aktuelle HTML-Datei befindet.

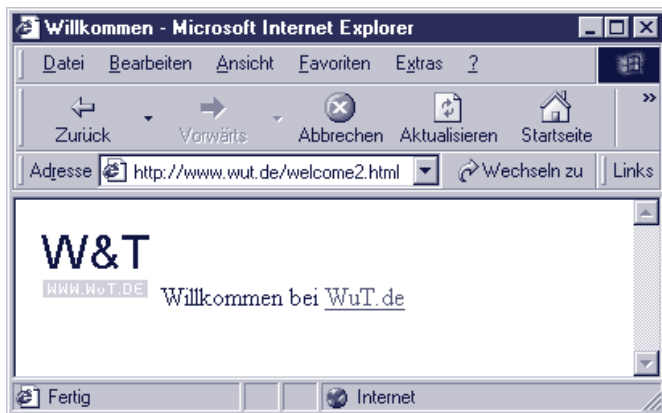
Darstellung von multimedialen Inhalten

Wie bereits angesprochen, erlaubt HTML die Darstellung von Inhalten, die nicht Bestandteil des HTML-Dokuments sind, sondern von anderer Stelle nachgeladen werden. Zur Einbindung von Bilddateien stellt HTML z.B. das ``-Tag (*img* für Image) zur Verfügung, wobei über das Attribut *src* Namen und Quelle der Bilddatei angegeben werden.

Wir erweitern unser HTML-Dokument um einen Grafik:

```
<body bgcolor="#FFFFFF">  
      
    Willkommen bei <a href="http://www.wut.de/index.html">WuT.de</a>  
</body>
```

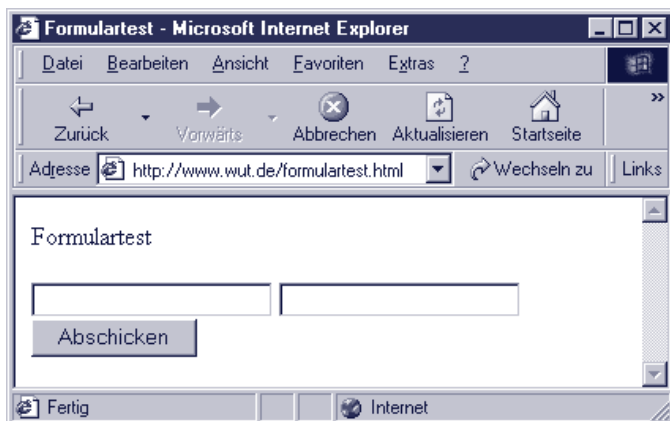
Nun wird neben dem Text ein Logo im GIF-Format dargestellt, das aus dem Verzeichnis *kpics* auf dem Webserver von W&T geladen wird. Der Pfad der Bilddatei kann wie auch beim Hyperlink absolut oder relativ angegeben werden.



HTML ist eine reine Darstellungssprache, die eine statische Anzeige im Browser generiert. Aber wie sieht es aus, wenn der Anwender Informationen an den WWW-Server zurückgeben möchte?

Als Lösung bietet HTML die Möglichkeit, Formulare anzuzeigen, die vom Anwender ausgefüllt werden können. Die so eingegebenen Informationen können durch Anklicken eines sogenannten „Submit-Buttons“ vom Browser zum WWW-Server gesendet werden.

Hier ein kurzes Beispiel:



Im HTML-Code sieht das so aus:

```
<html>
  <head>
    <title>Formulartest</title>
  </head>
  <body bgcolor="#FFFFFF">
    Formulartest
    <form method="post" action="Formularauswertung.cgi" name="FORMULAR1">
      <input type="text" name="EINGABEFELD1">
      <input type="text" name="EINGABEFELD2">
      <input type="submit" name="submit" value="Abschicken">
    </form>
  </body>
</html>
```

Sämtliche zum Formular gehörenden Elemente sind zwischen dem einleitenden *<form>*-Tag und dem abschließenden *</form>*-Tag zu finden.

Die Attribute des Form-Tags sind:

| | |
|---------------|--|
| method | gibt an, wie HTTP die Eingaben an den WWW-Server übergibt. |
| action | legt fest, an welchen Prozeß auf dem Server die Eingaben übergeben werden. |
| name | kann willkürlich vergeben werden und zeigt dem Prozeß auf dem Server, von welchem Formular die Eingaben stammen (ein Prozess kann ggf. mehrere Formulare auswerten). |

Die Eingabeelemente selbst werden über das *<input>*-Tag festgelegt, wobei das Attribut *type* angibt, um welche Art von Eingabeelement es sich jeweils handelt. Mögliche Attribute sind hier:

| | |
|-----------------|---|
| text | Texteingabefeld |
| checkbox | Ankreuzkästchen |
| radio | Optionsbutton |
| submit | Button zum Abschicken oder Zurücksetzen des Formulars |

Über das Attribut *name* kann dem Element ein eindeutiger Name (vergleichbar mit einem Variablennamen) gegeben werden; mit dem Attribut *value* kann ihm ein Anfangswert zugeteilt werden.

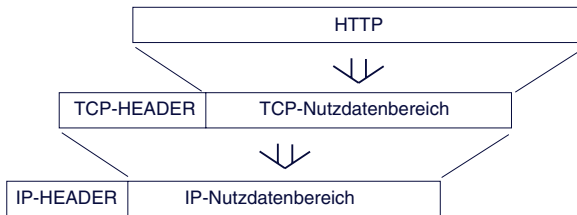
Die Angabe `<input type="text" name="EINGABEFELD1" value="test1">` würde z.B. dazu führen, dass beim Öffnen des Formulars im Browser schon der Text *test1* im ersten Eingabefeld stehen würde.

Was mit den per Formular übergebenen Informationen geschieht – ob der Anwender eine Rückmeldung erhält und wie diese aussieht –, bestimmt allein der Prozeß auf dem WWW-Server, der die Informationen entgegennimmt und auswertet.

Wie bereits angesprochen, möchten wir hier nicht bis auf das letzte Detail von HTML eingehen. Wer Webseiten erstellen möchte, sollte sich auf jeden Fall eingehender mit diesem Thema beschäftigen. Eine hervorragende Quelle für weitere Informationen zu HTML findet man unter <http://www.teamone.de/selfhtml/selfhtml.htm>; sehr nützlich ist selbstverständlich auch die Website des W3-Konsortiums (<http://www.w3.org>), das in Sachen HTML als normierende Körperschaft fungiert.

HTTP – Hypertext Transfer Protocol

Durch die rasante Zunahme von WWW-Nutzern ist HTTP heute das mit Abstand meist genutzte Protokoll im Internet. HTTP setzt auf TCP als Basisprotokoll auf, wobei in aller Regel der TCP-Port 80 genutzt wird (abweichende Ports sind möglich, müssen aber explizit im URL angegeben werden).



Die Anforderung und Übertragung einer Webseite erfolgt in vier Schritten:

1. Auflösen des angegebenen Host und Domainnamens in eine IP-Adresse

Der TCP/IP-Stack startet eine DNS-Anfrage um die IP-Adresse des gewünschten Servers zu ermitteln.

2. Aufbau der TCP-Verbindung

Zur Erinnerung: Bei einer TCP-Verbindung gilt das Client-Server-Prinzip. Bei HTTP übernimmt der Browser die Rolle des Client und stellt die TCP-Verbindung zum angegebenen WWW-Server her.

3. Senden der HTTP-Anforderung

Nach erfolgreichem Aufbau der TCP-Verbindung fordert der Browser die gewünschte Webseite beim WWW-Server an. An dieser Stelle beginnt das eigentliche HTTP-Protokoll: Der Browser sendet das *Get*-Kommando mit den erforderlichen Parametern zum WWW-Server.

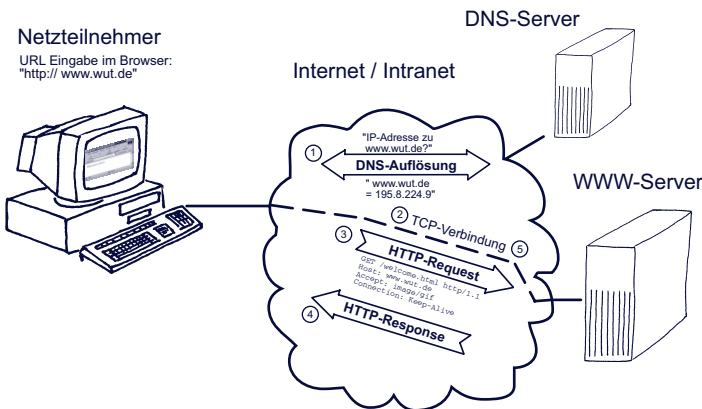
4. Senden der angeforderten Webseite

Der WWW-Server sendet erst eine HTTP-Bestätigung und dann die Webseite selbst.

5. Beenden der TCP-Verbindung durch den WWW-Server

Eine Besonderheit bei HTTP ist, dass die TCP-Verbindung nicht wie sonst üblich durch den Client, sondern durch den Server abgebaut wird. Dafür gibt es zwei Gründe:

- Der WWW-Server signalisiert dem Browser auf einfache Art und Weise, dass die Übertragung abgeschlossen ist. Eine empfangene Webseite wird im Browser deshalb auch erst dann angezeigt, wenn die TCP-Verbindung beendet ist.
- WWW-Server müssen eine Vielzahl von TCP-Verbindungen gleichzeitig bedienen. Dabei verlangt jede offene Verbindung dem Server ein gewisses Maß an Leistung ab. Um die Verbindungszeiten so kurz wie möglich zu halten, baut der Server die Verbindung einfach ab, sobald alle angeforderten Daten übertragen wurden.



Die wichtigsten HTTP-Kommandos und Parameter

Wie bereits angesprochen basiert auch HTTP auf dem Client-Server-Prinzip: Der Browser als Client kann durch das Senden bestimmter Kommandos die Kommunikation steuern.

Das GET-Kommando

Das mit Abstand am häufigsten verwendete Kommando ist die **GET-Anfrage**, die jeden Aufruf einer Webseite einleitet. GET fordert den HTTP-Server auf, ein Dokument oder Element zu senden und ist damit das wichtigste Kommando.

Für den Einsatz von GET sind einige Parameter nötig; man spricht auch von einer Kommandozeile (engl. Request Line).

```
GET /pfadname/filename http-Version
```

Weitere Parameter können jeweils als neue Zeile mitgesendet werden. Diese angehängten Parameter werden auch als „Header“ bezeichnet.

| | |
|-------------------|--|
| Host | Hostname (nur bei HTTP1.1 nötig). |
| Accept | <p>gibt an welche Dateiformate der Browser verarbeiten kann</p> <p>Mit <i>Accept: image/gif</i> gibt der Browser z.B. bekannt, dass er Bilder im GIF-Format anzeigen kann.</p> |
| Connection | <p>über diesen Parameter (<i>Connection: Keep-Alive</i>) kann vom Browser vorgegeben werden, ob die TCP-Verbindung zum Nachladen anderer Elemente offengehalten wird.</p> |

Eine Vielzahl weiterer Parameter sind in der RFC2616 beschrieben, die unter <http://www.w3.org/Protocols/rfc2616/rfc2616.html> eingesehen werden kann.

Ein typisches Get-Kommando könnte etwa so aussehen:

```
GET /welcome.html http/1.1
Host: www.wut.de
Accept: image/gif
Connection: Keep-Alive
```

Als Antwort sendet der HTTP-Server eine Statuszeile, auf die ein Header (diesmal mit Parametern des Servers) folgt. Getrennt durch eine Leerzeile <CR LF CR LF> wird das angeforderte Element übermittelt.

| | |
|---|-------------|
| HTTP/1.1 200 OK | Statuszeile |
| Date: Thu, 15 Mar 2001 11:33:41GMT | |
| Server: Apache/1.3.4 (Unix) PHP/3.0.6 | |
| Last-Modified: Thu 15 Mar 2001 11:32:32 GMT | |
| ... | |
| ... | Header |
| Keep-Alive: timeout=15 | |
| Connection: Keep-Alive | |
| Content-Type: text/html | |
| <html> | |
| ... | HTML-Seite |
| </html> | |

Die Statuszeile umfasst die vom Server unterstützte HTTP-Version, eine Fehlercode-Nummer und einen Kommentar. Im Header zeigt der Server unterstützte Verbindungseigenschaften und Daten an.

Das POST-Kommando

Das Gegenstück zu GET ist das POST-Kommando. POST erlaubt dem Browser, Informationen an den HTTP-Server zu übergeben.

Der klassische Einsatz für das POST-Kommando ist die Übergabe von Formulareinträgen aus einer HTML-Seite. Im Kern ist der Aufbau der POST-Anforderung identisch mit der von GET. Nach den Parametern steht eine Leerzeile <CR LF CR LF>, der die zu übergebenden Informationen folgen. Enthält eine POST-Anforderung mehrere Einzelinformationen, werden diese durch ein „&“ voneinander getrennt. Als *filename* muss in der ersten Zeile der POST-Anforderung ein auf dem Server verfügbarer Prozeß angegeben werden, der die Informationen entgegennehmen und verarbeiten kann.

Für das im HTML-Abschnitt gezeigte Formulartest-Formular könnte die POST-Anforderung folgendermaßen aussehen; der bislang nicht besprochene Parameter *Referer* stellt hier einen Bezug zu der ursprünglich geladenen Formular-Seite her:


```
POST /Formularauswertung.cgi HTTP/1.1
Accept: image/gif, image/jpeg
Referer: http://172.16.232.145/formulartest.html
Host: 172.16.232.145
Connection: Keep-Alive
```

```
EINGABEFELD1=test1&EINGABEFELD2=test2&submit=Abschicken
```



Tipp: Die meisten Internet Provider bieten sogenannte „CGI-Scripts“ (Programme auf dem HTTP-Server) an, die Formularangaben entgegennehmen und als E-Mail an eine beliebige Adresse weiterleiten. So kann man seinen Kunden z.B. die Gelegenheit geben, direkt von einer Webseite aus, eine Bestellung oder Anfrage zu verschicken.

Das HEAD-Kommando

Als drittes Kommando sei hier der Vollständigkeit halber noch eine Variante von GET genannt. Das **HEAD**-Kommando arbeitet wie das GET-Kommando, doch der HTTP-Server gibt nur die Statuszeile und den Header, nicht aber das angeforderte Element selbst zurück.

Es wird fast ausschließlich zu Testzwecken und von Suchmaschinen genutzt, die über die resultierende Meldung (Fehlercode) die Existenz einer Seite überprüfen können.

HTTP-Versionen

HTTP wurde seit der Einführung des WWW mehrfach weiterentwickelt und kommt heute in drei Versionen vor:

- HTTP 0.9** in 1989 erstmalig vorgestellt und seit dem genutzt, aber nie spezifiziert
- HTTP 1.0** erst 1996 wurde HTTP in der Version 1.0 durch die RFC 1945 spezifiziert, die weitestgehend mit HTTP0.9 identisch ist
- HTTP 1.1** wurde 1997 (RFC 2068) eingeführt und ist seit 1999 (RCF 2616) in überarbeiteter Form im Einsatz.

Alle heute erhältlichen Browser unterstützen standardmäßig HTTP1.1, können aber auch problemlos mit Servern zusammenarbeiten, die HTTP0.9 oder HTTP1.0 verwenden.

Die wohl grundlegendste Änderung in HTTP1.1 liegt darin, dass die für die Übertragung des HTML-Dokuments aufgebaute TCP-Verbindung, auch für das Nachladen weiterer Elemente weitergenutzt wird. HTTP1.0 bzw. 0.9 haben für jedes Element eine separate TCP-Verbindung aufgebaut.

Eine persistente Verbindung wie in 1.1 erhöht den Datendurchsatz, da die Zeiten für Verbindungsaufbau und -abbau entfallen.

Als weitere Neuerung in der Version 1.1 kann ein HTTP-Server mit nur einer IP-Adresse Anfragen an verschiedene Hostadressen verarbeiten. Trägt der Anwender im Browser als URL z.B. *http://www.wut.de* ein, fragt der PC beim DNS-Server die zugehörige IP-Adresse nach.

Der Browser öffnet die TCP-Verbindung und sendet das GET-Kommando. Um auf einem HTTP-Server die Internet-Auftritte mehrerer Anbieter verwalten zu können, wurde mit *Host* ein zusätzlicher Parameter zum GET-Kommando eingeführt, der dem Server zusammen mit einer GET-Anfrage auch den Hostnamen übermittelt (z.B. *Host: http://www.wut.de*). Dank dieses zusätzlichen Parameters kann der HTTP-Server über die GET-Anfrage erkennen, welchem Host die TCP-Verbindung gilt.

Interaktivität im WWW

Neben der rein statischen Darstellung von Informationen (Web-Seiten) gibt es verschiedene Möglichkeiten, vom Browser Aktionen auszulösen und Elemente dynamisch anzuzeigen.

Dazu ist auf jeden Fall ein Programm, bzw. ein Prozess nötig, der z.B. Eingaben vom Anwender entgegennimmt und entsprechende Reaktionen auslöst.

Unterschieden wird zwischen Programmen, die auf dem WWW-Server aktiv sind und solchen die im Browser, also auf dem lokalen Rechner ablaufen. Auch eine Kombination von beidem ist oft zu finden.

Interaktivität durch Programme die auf dem Server ablaufen

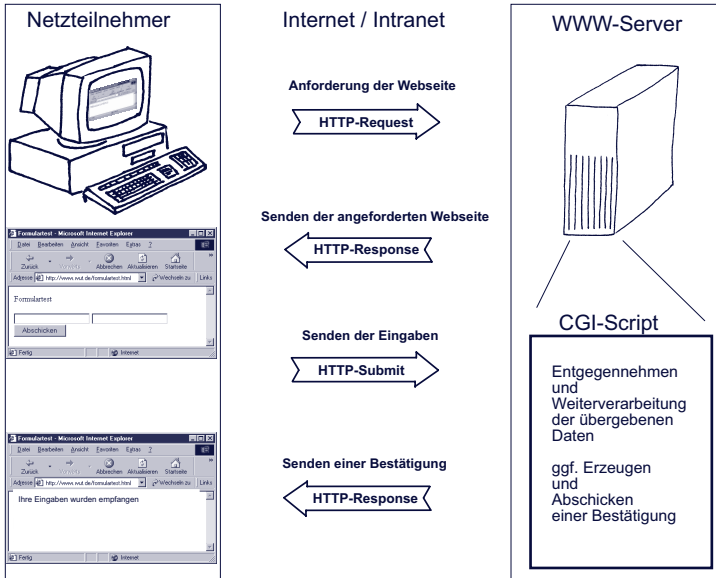
CGI - Common Gateway Interface

Der Einsatz von CGI-Scripten ist das zur Zeit meistgenutzte Verfahren, im Browser interaktive Inhalte anzuzeigen, bzw. Aktionen auszulösen.

Über CGI können vom Browser aus Programme auf dem WWW-Server ausgeführt werden.

Über einen Hyperlink, einen Submit-Button oder direkte Eingabe des URL wird das entsprechende Programm aufgerufen und es werden ggf. die nötigen Parameter übergeben.

Ein klassisches Beispiel sind HTML-Formulare, die vom Anwender ausgefüllt werden. Klickt der Anwender den Submit-Button (Abschicken) werden die Eingaben via http mit Hilfe des POST-Kommandos an den WWW-Server übergeben. Das angegebene CGI-Script wird gestartet und verarbeitet die Eingaben weiter.



Weitere mögliche Anwendungen sind Besucherzähler, Gästebücher, Diskussionsforen, Datenbankzugriffe oder Suchmaschinen.

CGI-Skripte können grundsätzlich in allen gängigen Programmiersprachen erstellt werden. Wichtig ist, dass der WWW-Server die gewählte Sprache unterstützt.

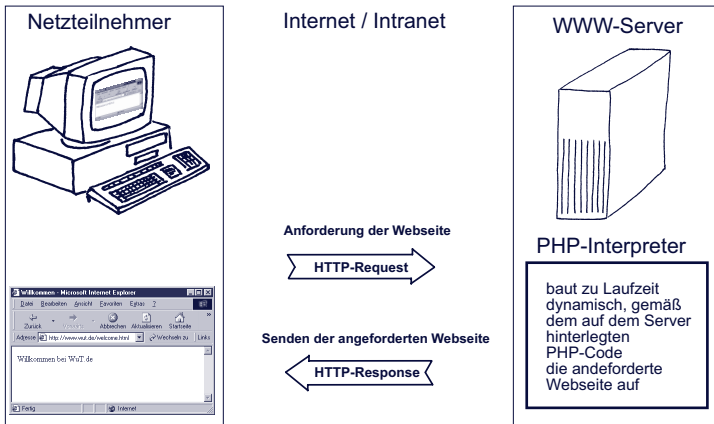
In der Praxis hat sich Perl für die Erstellung von CGI-Skripten durchgesetzt.

PHP

Auch PHP erlaubt das Ausführen von Programmen auf einem WWW-Server. PHP ist eine Interpretersprache, deren Quelltext in eine HTML-Seite eingebunden ist, die auf dem WWW-Server liegt. Dabei können statische Inhalte der Seite im HTML-Format definiert werden, wogegen veränderbare Inhalte durch PHP-Quellcode eingebracht werden. PHP kann auch auf andere Ressourcen auf dem Server, wie z.B. Datenbanken zugreifen.

Bei Anforderung der entsprechenden Seite durch den Browser, wird der in der Seite integrierte PHP-Code auf dem Server, vom PHP-Interpreter ausgewertet.

Der PHP-Interpreter erzeugt individuell eine Seite in HTML-Code. Die so entstandene Webseite wird dann vom Server via http zum Browser gesendet.



So bleibt der PHP-Quellcode für den Anwender unsichtbar.

Beim Onlineshopping könnte man zum Beispiel zu den offerierten Artikeln dynamisch Lagerstückzahlen, Lieferzeiten und Preise aus einer Warenwirtschaftsanwendung via PHP in die Webseite einbringen.

Das bedingt natürlich, dass auf dem Server ein PHP-Interpreter aktiv ist.

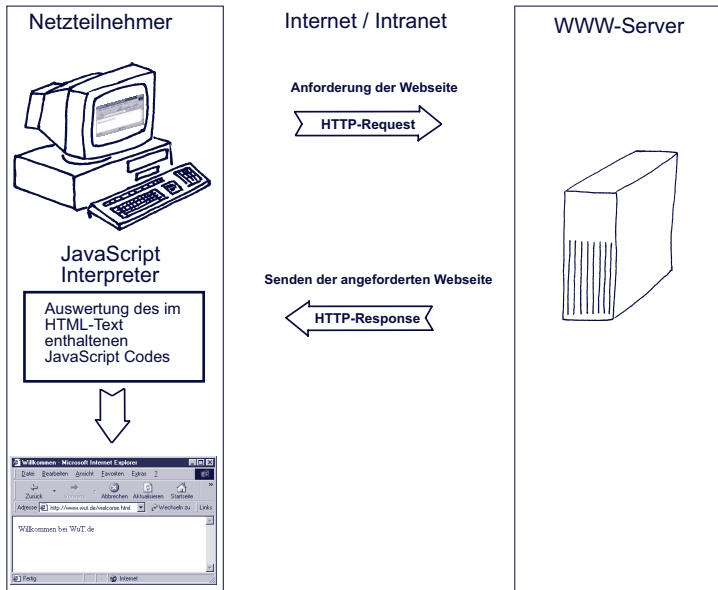
PHP wird z.Zt. in den Versionen PHP3 und PHP4 verwendet.

Programme die im Browser ausgeführt werden.

JavaScript

Bei JavaScript wird der Quellcode in den HTML-Text der Seite eingebunden. Der JavaScript Code wird mit dem `<SCRIPT language="JavaScript">` Tag gekennzeichnet und beim Laden

einer Webseite vom Browser erkannt, interpretiert und ausgeführt.



Mit JavaScript können z.B. individuelle Anpassungen der angezeigten Inhalte einer Web-Seite vorgenommen werden. Auch Benutzereingaben lassen sich überprüfen, bevor sie zum WWW-Server weitergeleitet werden.

Ein Beispiel:

Der folgende Code wertet aus, ob eine Web-Seite über die „com“ Domain oder die „de“ Domain aufgerufen wurde und stellt sich entsprechend englisch oder deutsch dar.

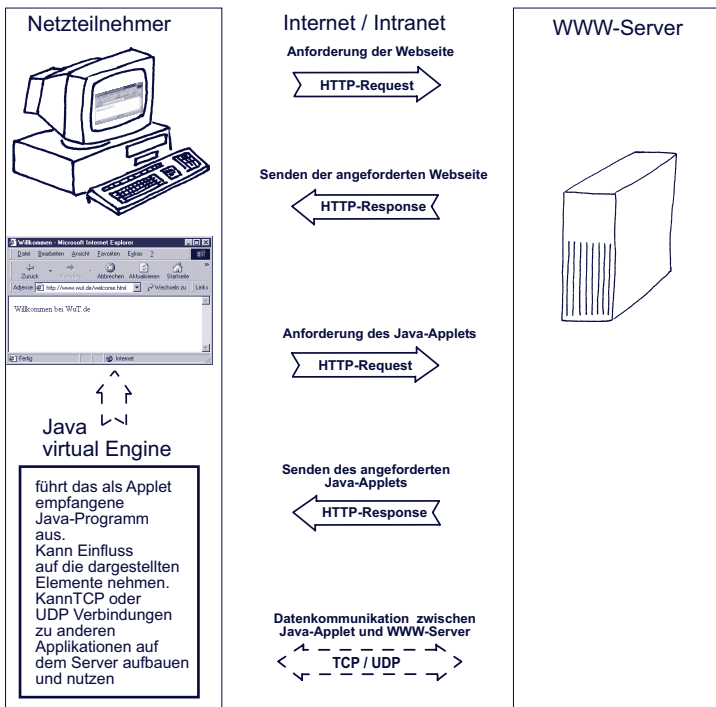
```
<HTML>
<HEAD>
  <TITLE>urltest</TITLE>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
</HEAD>
<BODY>
  <SCRIPT LANGUAGE="JavaScript"><!--
    if (location.hostname == "www.web-io.com") document.write("welcome at WuT");
    else document.write("willkommen bei WuT");
```

```
//--></SCRIPT>
</BODY>
</HTML>
```

Java Applets

Hier handelt es sich um kompilierte Programme, die in der Programmiersprache Java erstellt wurden. Java Applets werden, ähnlich wie grafische Elemente, zusätzlich zum HTML-Text geladen und im Browser ausgeführt.

Mit Java Applets können auch komplexere Aktionen, wie Netzwerkzugriffe auf TCP- und UDP-Ebene realisiert werden.



Aus Sicherheitsgründen ist allerdings nur die Kommunikation mit dem Server möglich, von dem das Applet geladen wurde. Auch auf dem lokalen Rechner des Anwenders ist der Zugriff auf Elemente und Funktionen des Browsers beschränkt. Ein Zu-

griff auf die Festplatte des eigenen Rechners ist zum Beispiel nicht möglich.

Ein Beispiel für den Einsatz von Java Applets ist das W&T Web-Thermometer. Vom Web-Thermometer wird ein Applet zur Verfügung gestellt, dass einmal im Browser gestartet, in regelmäßigen Abständen die aktuelle Temperatur abfragt und in der Webseite darstellt, von der es aufgerufen wurde.

Im HTML Code werden Applets über das Applet-Tag eingebunden, wobei mit dem Parameter *code*= der Name des Applets angegeben wird und unter *codebase*= der Host von dem das Applet geladen wird.

```
<html>
<head>
  <title>Schaltschranktemperatur</title>
</head>
<body bgcolor="#FFFFFF">
  <p><applet code="A.class" codebase = "http://172.16.232.152/" ></applet> </p>
</body>
</html>
```

im Browser sieht das dann so aus:



E-Mail

Die Möglichkeit, elektronische Post in wenigen Sekunden von einem Ende der Welt zum anderen verschicken zu können, ist sicherlich einer der Hauptgründe für die rasante Ausbreitung des Internet.

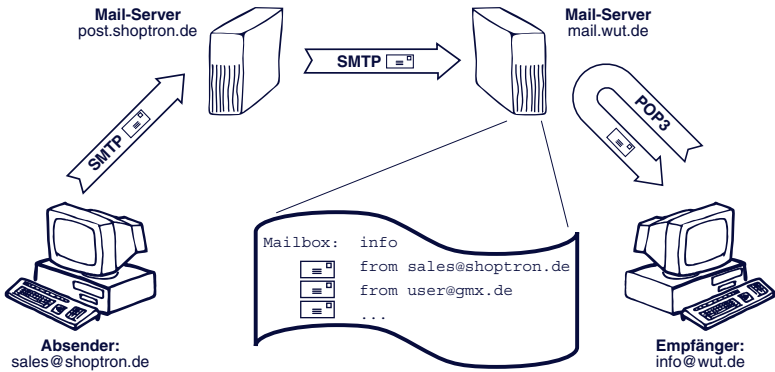
Im Gegensatz zu den meisten anderen Anwendungen im Internet ist das Versenden von E-Mail ein Dienst, bei dem keine direkte Verbindung zwischen Sender und Empfänger besteht. Das klingt zunächst verwirrend, ist aber sinnvoll, da sonst der Austausch von E-Mail nur möglich wäre, wenn Versender und Empfänger gleichzeitig im Netz aktiv sind.

Um eine zeitliche Unabhängigkeit zu gewährleisten, benötigt der E-Mail-Empfänger eine Mailbox (Postfach) auf einem Mail - Server, in der eingehende Nachrichten zunächst abgelegt werden.

Eine E-Mail-Adresse setzt sich immer aus dem Postfachnamen und der Zieldomain zusammen; als Trennzeichen steht das „@“ (engl. „at“, gesprochen „ätt“) zwischen diesen beiden Bestandteilen. Ein Beispiel: *info@wut.de* bezeichnet das Info-Postfach auf dem Mailserver von Wiesemann & Theis.

Der Weg einer E-Mail vom Versender zum Empfänger besteht aus zwei Teilabschnitten, auf denen der Transport über unterschiedliche Protokolle geregelt wird:

- vom Rechner des Absenders bis zum Postfach des Empfängers wird das SMTP-Protokoll benutzt,
- vom Postfach des Empfängers bis zum Rechner des Empfängers wird das POP3-Protokoll benutzt.



Aufbau einer E-Mail

Eine E-Mail setzt sich aus dem Nachrichtenkopf und der eigentlichen Nachricht zusammen. Diesen Kopf kann man mit einem Briefumschlag vergleichen, der Felder für Absender, Empfänger, Datum, Betreff und einige weitere Informationen enthält.

Hier die wichtigsten Felder im Überblick:

Die folgenden vier Felder bilden einen Minimalkopf und müssen auf jeden Fall enthalten sein.

| Feld | Funktion |
|----------|--|
| FROM | E-Mail-Adresse des Verfassers |
| TO | E-Mail-Adresse des Empfängers |
| DATE | Datum und Uhrzeit Hinweis: Die Uhrzeit kann willkürlich eingetragen werden und ist in aller Regel die Ortszeit des Absenders. |
| SUBJECT | Text der Betreffzeile |
| RECEIVED | Das Feld RECEIVED stellt eine Besonderheit dar, denn es wird nicht bei Erstellung der E-Mail angelegt. Jeder auf dem Weg der E-Mail liegende Mail-Router fügt ein RECEIVED-Feld ein und hinterlässt auf diese Weise einen "Durchgangsstempel" mit Datum und Uhrzeit. |

Die Verwendung der im folgenden genannten Felder ist optional.

| Feld | Funktion |
|--------------|--|
| SENDER | E-Mail-Adresse des Absenders (in aller Regel identisch mit Eintrag unter FROM) |
| REPLY-TO | E-Mail-Adresse, an die der Empfänger im Bedarfsfall antworten soll. Wichtig, wenn E-Mails von einem Embedded-System wie dem W&T IO-Mailer automatisiert verschickt werden. Als Antwortadresse könnte in diesem Fall z.B die E-Mail-Adresse des Administrators eingetragen sein. |
| CC | E-Mail-Adresse eines weiteren Empfängers, der einen "Durchschlag" (CC = "Carbon Copy") der Nachricht erhält. |
| BCC | E-Mail-Adresse eines weiteren Empfängers, die für alle anderen Empfänger aber unsichtbar bleibt (BCC = "Blind Carbon Copy"). |
| MESSAGE-ID | Eindeutige Identifikation einer E-Mail, die von der Mailsoftware willkürlich vergeben wird. |
| X-"MEINFELD" | Durch Voranstellen von "X-" können eigene Felder erzeugt werden. |

Bei einigen Feldern ist eine RESENT-Variante möglich, die dann zum Tragen kommt, wenn es sich um eine vom ursprünglichen Empfänger weitergeleitete E-Mail handelt.

Der formale Aufbau von Nachrichtenkopf und Feldern muss den folgenden Konventionen genügen:

- Nach dem Feldnamen steht ein Doppelpunkt; danach folgt der jeweilige Parameter.
- Jedes Feld steht in einer eigenen Zeile, die mit <CR LF> (Carriage Return Line Feed; hex 0D 0A) endet.
- Nachrichtenkopf und Körper werden durch eine zusätzliche Leerzeile <CR LF> getrennt.
- Der Nachrichtenkörper selbst enthält nur den zu übermittelnden Text bzw. weitere eingefügte Dateien. Das Ende der Nachricht wird durch <CR LF . CR LF> (hex 0D 0A 2E 0D 0A) gekennzeichnet.
- Sowohl Kopf als auch Nachrichtenkörper bestehen ausschließlich aus 7-Bit-ASCII-Zeichen. Deshalb können auch alle Steuerinformationen als Klartext übertragen werden.

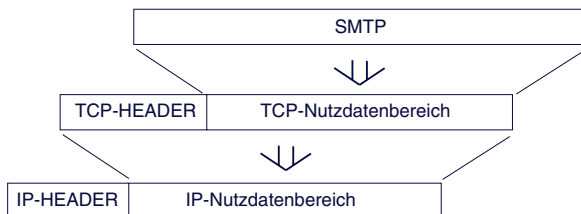
MIME – Multipurpose Internet Mail Extensions

Um auch binäre Daten (8-Bit-Format) via E-Mail verschicken zu können, werden diese vor dem Einbinden in den Nachrichtenkörper nach dem „MIME-Standard“ in das 7-Bit-Format codiert und beim Empfänger wieder decodiert. Da die Verarbeitung binärer Daten von heutigen E-Mail-Programmen automatisch übernommen wird, verzichten wir an dieser Stelle auf eine detaillierte Erklärung der „MIME-Codierung“.

SMTP – Simple Mail Transfer Protocol

SMTP regelt den Versand von E-Mails vom Mail-Client zum Mailserver (SMTP-Server). Der Mail-Client kann dabei entweder der ursprüngliche Versender oder ein auf dem Weg liegender Mail-Router sein. Mail-Router kommen zum Einsatz, wenn die E-Mail auf ihrem Weg über mehrere Domains weitergereicht wird. Häufig findet man für Mail-Router auch die Bezeichnung *MTA* (Mail-Transfer-Agent).

Für jedes Teilstück, das eine E-Mail zurücklegt, wird eine eigene TCP-Verbindung aufgebaut. SMTP setzt auf diese TCP-Verbindung auf, wobei der TCP-Port 25 genutzt wird.



SMTP stellt einige Kommandos (z.B. Login, Angabe des Absenders, Angabe des Empfängers ...) zur Verfügung. Jedes SMTP-Kommando wird einzeln vom SMTP-Server quittiert. Die eigentliche E-Mail wird komplett mit Kopf und Körper gesendet und dann erst vom SMTP-Server quittiert. Wenn keine weiteren E-Mails zum Versand anstehen, wird auch die TCP-Verbindung wieder abgebaut.

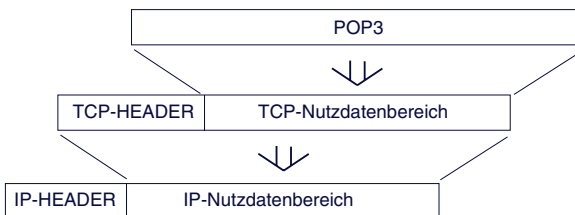
Hat die E-Mail den Ziel-Mailserver erreicht, wird sie im Postfach des Empfängers abgelegt und bleibt dort so lange liegen, bis sie vom Empfänger abgeholt wird.

POP3 – Post Office Protocol Version 3

Um eingegangene E-Mails aus dem Postfach auf dem Mailserver abzuholen, wird in den meisten Fällen das POP3-Protokoll benutzt. Der Empfänger wird über eingehende E-Mails nicht informiert. Er muss sein Postfach selbständig auf eingegangene E-Mails überprüfen und kann diese zu einem beliebigen Zeitpunkt abholen.

Die meisten der heute genutzten E-Mail-Programme überprüfen bei Start zunächst automatisch das Postfach des Nutzers auf eingegangene Mail. Viele E-Mail-Programme bieten darüber hinaus die Möglichkeit, ein Intervall vorzugeben, in dem das Postfach zyklisch geprüft wird. Typische Nutzer, die die meiste Zeit des Tages „offline“ sind, erhalten ihre E-Mails ohnehin nur dann, wenn Sie sich beim Provider eingewählt haben. Doch bei Computern mit permanentem Internetzugang ist die zyklische Abfrage durchaus sinnvoll: Der Nutzer ist hier ständig online und erhält seine E-Mails mit nur geringer Verzögerung – quasi in Echtzeit.

Auch das POP3-Protokoll setzt auf eine TCP-Verbindung auf und ist nichts anderes als ein Klartextdialog.



POP3 nutzt die TCP-Portnummer 110. Wie bei SMTP beginnt der Dialog auch hier mit einem Login. Bei POP3 muss sich der Empfänger allerdings in zwei Schritten anmelden: mit Nutzernamen

und mit Passwort. Nach erfolgreichem Login stellt POP3 einige Kommandos zur Verfügung, mit denen eingegangene Nachrichten aufgelistet, abgeholt oder gelöscht werden können.

Heute wird der Nutzer mit SMTP und POP3 nur noch in geringem Maße konfrontiert: Er muss lediglich beim Einrichten der Mailsoftware den Namen des POP3- und SMTP-Servers angeben – das Abwickeln der Protokolle selbst wird unsichtbar im Hintergrund vom Mailprogramm übernommen.

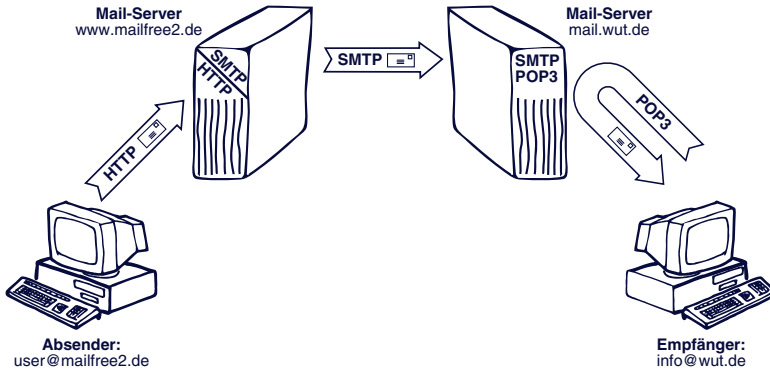
Der Vollständigkeit halber sei noch erwähnt, dass es neben dem POP3-Protokoll noch die Protokolle POP2 und POP1 (beides Vorläufer von POP3) und IMAP4 gibt, die ebenfalls zum Abholen von E-Mails entwickelt wurden. Diese Protokolle konnten sich in der Praxis aber noch nicht durchsetzen oder wurden von POP3 verdrängt.

E-Mail über HTTP senden und empfangen

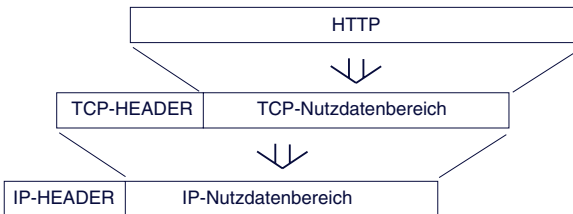
Mit der zunehmenden Nutzung von E-Mail gibt es immer mehr Freemail-Anbieter, die auf Ihrem Mailserver kostenlos Postfächer zur Verfügung stellen. Diese Dienstleistung, die jeder nutzen kann, wird in aller Regel über Werbung finanziert.

Um Raum zur Einblendung von Werbung zu schaffen, geben die meisten Freemail-Anbieter dem Nutzer die Möglichkeit, das Senden und Abrufen von E-Mails bequem über HTTP im Browser abzuwickeln, der selbstverständlich durch Werbeanzeigen bereichert ist. Hierzu stehen dem Nutzer entsprechende HTML-Formulare zur Verfügung.

Um die E-Mail-Abwicklung über HTTP zu ermöglichen, muss der Freemail-Anbieter eine spezielle Mailserver-Kombination betreiben, die zur Nutzerseite als Webserver, zur anderen Seite als SMTP-Server arbeitet. Der Weg einer E-Mail sieht hier folgendermaßen aus:



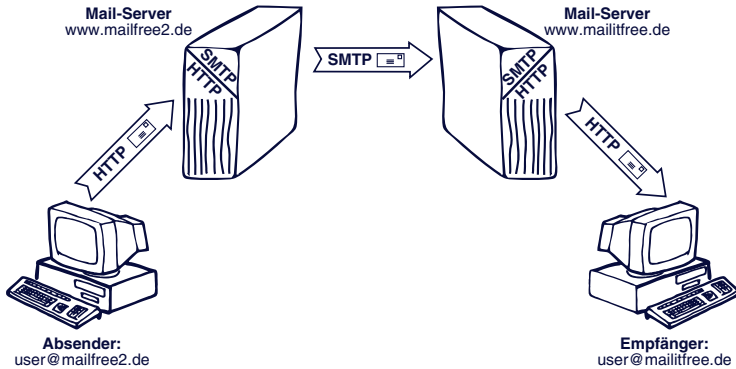
Zwischen dem Rechner des Absenders und dem Server des Freemail-Anbieters wird das HTTP-Protokoll verwendet. Wie bei anderen HTTP-Anwendungen auch, wird auch hier die TCP-Portnummer 80 genutzt.



Zwischen den Mailservern selbst ändert sich nichts. Sie kommunizieren miteinander über das SMTP-Protokoll.

Zwischen dem Ziel-Mailserver und dem Rechner des Empfängers können zwei unterschiedliche Varianten zum Einsatz kommen:

- Hat der Empfänger ein Standard-Mailkonto, werden eingegangene Mails über POP3 abgeholt.
- Nutzt auch der Empfänger die Dienste eines Freemail-Anbieters, kommt hier ebenfalls HTTP zum Einsatz.



Wer seine E-Mail lieber über SMTP und POP3 versenden möchte, sollte bei der Wahl des Freemail-Anbieters unbedingt darauf achten, dass auch Zugangsmöglichkeiten über einen SMTP- bzw. POP3-Server vorhanden sind.

E-Mail und DNS

Auch beim Versenden von E-Mails wird auf IP-Ebene mit IP-Adressen gearbeitet. Die Namensauflösung bei E-Mail Adressen funktioniert vom Prinzip genauso wie bei normalen Netzteilnehmern auch. Natürlich wird dabei nicht die Adresse des E-Mail-Empfängers selbst aufgelöst, sondern lediglich die des Mailservers, auf dem der Empfänger sein Postfach hat.

Zur Erinnerung: Um Namen in Adressen aufzulösen, bedient sich der TCP/IP-Stack eines Resolver-Programms, das beim DNS-Server eine entsprechende Anfrage stellt.

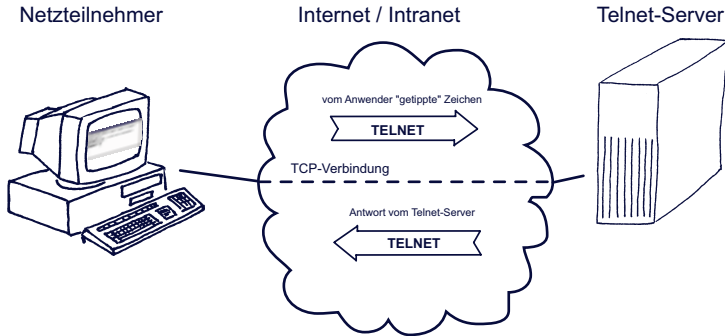
Nun ist der Hostname des Ziel-Mailservers aber nicht bekannt. Bekannt ist lediglich die Ziel-Domain, die ja in der E-Mail-Adresse hinter dem @-Zeichen steht. Um auch DNS-Anfragen nach Mailservern auflösen zu können, gibt es auf DNS-Servern spezielle Datensätze, in denen die zu einer Domain gehörenden Mailserver samt der zugehörigen IP-Adressen verzeichnet sind.

Das Resolver-Programm gibt also bei der Anfrage nur den Ziel-
Domainnamen an und teilt zudem mit, dass es sich bei dem
gesuchten Netzteilnehmer um einen Mailserver handelt. Der
DNS-Server ermittelt die gesuchte IP-Adresse und gibt sie an
das Resolver-Programm zurück.

Der Postfachname selbst wird für die DNS-Anfrage gar nicht be-
nötigt. Er wird erst bei Eintreffen der Nachricht auf dem Ziel-
Mailserver ausgewertet, damit diese im richtigen Postfach ab-
gelegt werden kann.

Telnet - Terminal over Network

Einfach ausgedrückt ist Telnet ein Textfenster bzw. text-orientiertes Programm, über das ein anderer Rechner (Host) im Netzwerk, vom Anwender fernbedient werden kann.



Eine Telnet-Sitzung kann man sich vorstellen wie eine DOS-Box, allerdings werden die eingetippten Befehle auf dem entfernten Rechner ausgeführt.

```

Telnet - wlinux
Verbinden Bearbeiten Terminal ?
Welcome to SuSE Linux 6.3 (i386) - Kernel 2.2.13 (pts/3).

WLinux login: root
Password:
You have new mail in /var/spool/mail/root
Last login: Wed May  2 11:19:32 from FT1.wiesemann.de
Have a lot of fun...
WLinux:~ #
  
```

Dafür werden mehrere Elemente benötigt

Der Telnet Client

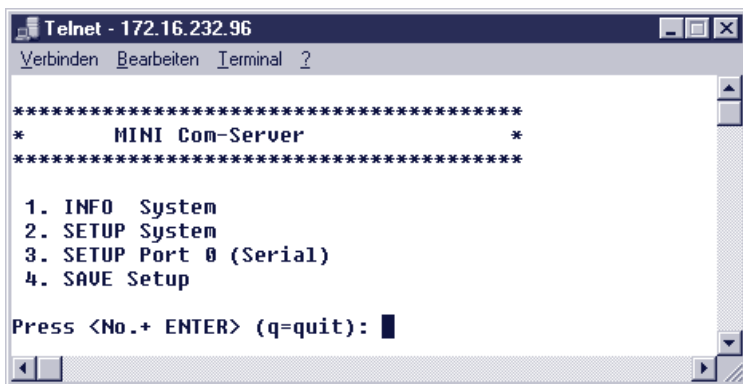
Alle modernen Betriebssysteme verfügen heute über ein Telnet-Clientprogramm.

Der Telnet-Client baut eine TCP-Verbindung zu einem Telnetserver auf, nimmt Tastatureingaben vom Anwender

entgegen, gibt sie an den Telnetserver weiter und stellt die vom Server gesendeten Zeichen auf dem Bildschirm dar.

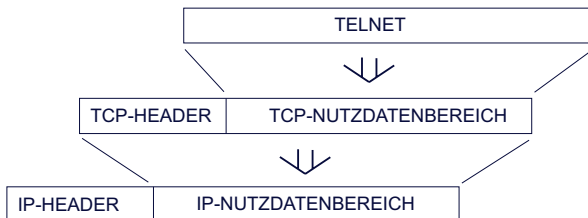
Der Telnet-Server

ist auf dem entfernten Rechner aktiv und gibt einem oder ggf. mehreren Nutzern die Gelegenheit sich dort „einzuloggen“. Damit ist der Telnet-Server (in Unix Systemen auch oft als Telnet-Daemon bezeichnet) das Bindeglied zwischen Netzwerkzugang via Telnet-Client und dem zu bedienenden Prozess. In seinem Ursprung wurde Telnet eingesetzt um einen Remote-Zugang zu UNIX-Betriebssystemen zu schaffen. Heute verfügen auch viele Embedded-Systeme wie Com-Server oder Printer-Server, Switches, Hubs und Router über einen Telnet-Server, der als Konfigurationszugang dient.



Das Telnet Protokoll

Auch Telnet setzt auf TCP als Basisprotokoll auf.



Hier bei wird, wenn vom Anwender nicht anders vorgegeben der Port 23 genutzt. Es kann aber auch jeder beliebige andere Port angegeben werden. Wichtig ist, dass auf dem gewählten Port ein Telnet-Server aktiv ist.

Das Telnet Protokoll übernimmt im wesentlichen drei Aufgaben:

1. Festlegung benutzter Zeichensätze und Steuercodes zur Cursorpositionierung
Als gemeinsame Basis für Client und Server wird hierzu der NVT-Standard „Network Virtual Terminal“ eingesetzt. NVT benutzt den 7Bit ASCII Zeichensatz und legt fest, welche Zeichen dargestellt werden und welche zur Steuerung und Positionierung genutzt werden.
2. Aushandeln und Einstellen von Verbindungsoptionen
Über die Festlegungen im NTV hinaus, kann Telnet von einer Vielzahl spezieller Funktionen Gebrauch machen. Das Telnet-Protokoll gibt Client und Server die Möglichkeit Verbindungsoptionen auszuhandeln. Zum Beispiel: ob der Server alle vom Client empfangenen Zeichen als Echo zurückgeben soll.

Hierzu werden Steuerzeichen benutzt, bei den das 8.Bit gesetzt ist, also Zeichen oberhalb 127 und damit außerhalb des NTV-Zeichensatzes.

3. Den Transport der Zeichen, die zwischen Client und Server ausgetauscht werden
Alle vom Anwender eingegebenen oder vom Server gesendeten Zeichen des NTV Zeichensatzes werden 1:1 in den Nutzdatenbereich eines TCP-Paketes gepackt und übers Netzwerk transportiert.
Die Einfachheit des Telnetprotokolls, sowie die Transparenz bei der Zeichenübertragung, haben Telnet auch zu einem beliebten Diagnosetool gemacht. So lassen sich Verbindungen zu http, SMTP oder POP3 Servern herstellen.

Es lässt sich zum Beispiel durch Eingabe der folgenden Zeile in einer Dosbox:

telnet <IP-Adresse eines Mail-Servers> 25

überprüfen, ob der SMTP-Server (Port25) arbeitet.

Ist der SMTP-Server aktiv, wird eine Begrüßungsmeldung zurückgegeben.



Durch konsequentes Eintippen des SMTP-Protokolls könnte man nun theoretisch per Telnet-Client E-Mails verschicken.

Auch andere einfache Protokolle wie http oder POP3 lassen sich via Telnet-Client per Hand nachvollziehen.

FTP - File Transfer Protocol

In einfachen Worten ausgedrückt, erlaubt FTP einem Anwender im Netzwerk den Zugriff auf das Datei-System, bzw. die Festplatte eines entfernten Rechners.

Der FTP-Client

FTP arbeitet nach dem Client/Server Prinzip. Ein FTP-Client ist heute Bestandteil jedes Betriebssystems. Unter Windows z.B. wird durch Eingabe des FTP-Befehls in einer Dosbox der FTP-Client gestartet.

Mit dem OPEN-Kommando, gefolgt von der IP-Adresse bzw. dem Hostnamen des FTP-Servers, wird die FTP Verbindung geöffnet und der Nutzer muss seinen Login-Namen und ein Passwort eingeben.

Nach erfolgreichem Login, sind je nach Zugriffsrecht unter anderem folgende Dateioperationen möglich:

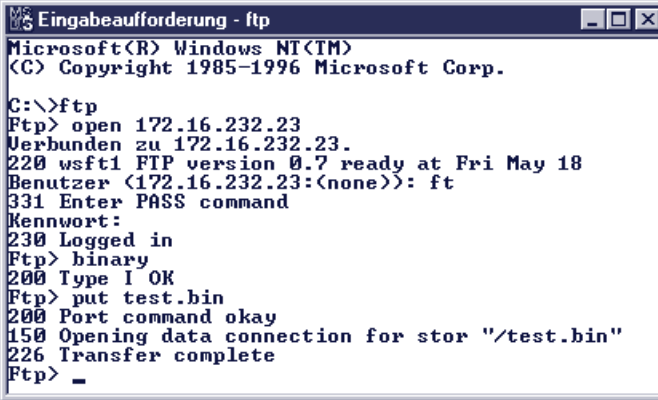
| | FTP Befehl |
|---|------------|
| Speicher von Dateien auf dem Server | PUT |
| Laden von Dateien vom Server | GET |
| Daten an eine bestehende Datei anhängen | APPEND |
| Löschen von Dateien auf dem Server | DELETE |
| Anzeigen des Verzeichnisinhaltes | DIR |

Eine Auflistung aller unterstützten Kommandos erhält man mit der Eingabe eines "?" hinter dem FTP-Prompt. Eine kurze Beschreibung der einzelnen Kommandos kann mit „? Kommando“ abgerufen werden.

Eine wichtige Eigenschaft von FTP ist die unterschiedliche Handhabung von Text- und Binärdateien. Um die gewünschte Betriebsart auszuwählen, stellt FTP zwei weitere Kommandos zur Verfügung:

| | |
|--------------------------------------|------------|
| | FTP Befehl |
| für die Übertragung von Textdateien | ASCII |
| für die Übertragung von Binärdateien | BINARY |
| (z.B. ausführbare Programmdateien) | |

Nach der Eingabe von FTP findet die Bedienung in einer Art Dialog statt, wie hier beispielhaft für das Speichern der Datei „test.bin“ auf dem Server „172.16.232.23“ gezeigt.



```
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

C:\>ftp
Ftp> open 172.16.232.23
Verbunden zu 172.16.232.23.
220 wsft1 FTP version 0.7 ready at Fri May 18
Benutzer (172.16.232.23:<none>): ft
331 Enter PASS command
Kennwort:
230 Logged in
Ftp> binary
200 Type I OK
Ftp> put test.bin
200 Port command okay
150 Opening data connection for stor "/test.bin"
226 Transfer complete
Ftp> _
```

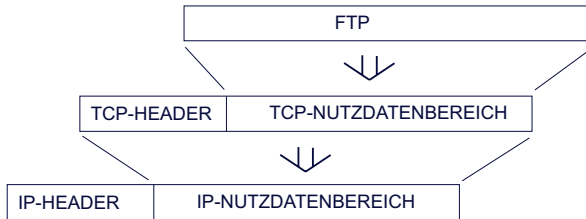
Je nach Betriebssystem können sowohl die Bedienung, als auch die Kommandos des FTP-Client variieren.

In Unix-Betriebssystemen ist außerdem strickt auf Groß- und Kleinschreibung zu achten.

Eine komfortablere Handhabung von FTP lässt sich durch den Einsatz von zugekauften FTP-Client Programmen mit grafischer Benutzeroberfläche erreichen.

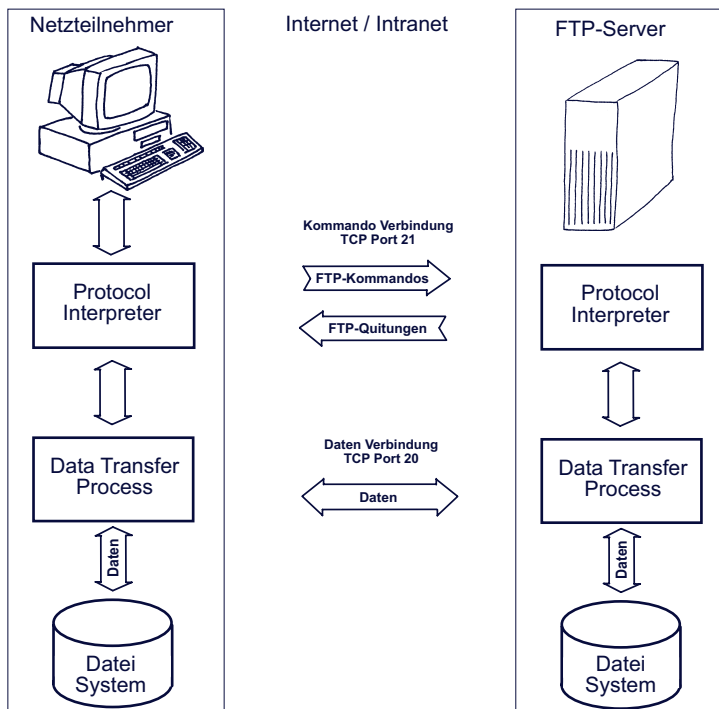
Das FTP-Protokoll

Als Basis-Protokoll setzt FTP, auf das verbindungsorientierte und gesicherte TCP auf.



Im Gegensatz zu anderen Interndiensten nutzt FTP aber zwei TCP-Verbindungen und damit zwei TCP-Ports

- Port 21 als Kommando-Verbindung
- Port 20 zur Übertragung von Dateien



Die Steuerung des Datei-Transfers wird zwischen Client und Server, wird über einen Kommandodialog gesteuert. Diesen Part wickeln die Protocol-Interpreter über die Kommando-

Verbindung ab. Die Kommando-Verbindung bleibt für die gesamte Dauer der FTP-Sitzung bestehen.

Der eigentliche Datei-Transfer erfolgt über die Daten-Verbindung, die vom Data Transfer Prozess für jede Dateioperation neu geöffnet wird.

Der Data Transfer Prozess ist dabei das Bindeglied zwischen Netzwerk und Dateisystem und wird vom Protocol-Interpreter gesteuert.

Der FTP-Server

Ein FTP-Server steht in der Regel nur bei Server-Betriebssystemen zur Verfügung und muss ggf. erst gestartet werden.

FTP-Server bieten zwei Zugriffsmöglichkeiten:

1. Nur eingetragene Nutzer haben Zugriff und können, je nach in einer Userliste festgehaltenem Zugriffsrecht, Dateioperationen ausführen.
2. Jeder Nutzer kann auf den Server zugreifen. Ein Login findet entweder gar nicht statt, oder es wird der Username „anonymous“ angegeben. Man spricht dann von Anonymous-FTP

Eines der Hauptanwendungen für FTP ist heute das aufspielen von HTML-Seiten auf WWW-Server, die zu diesem Zweck auch immer einen FTP-Zugang haben.

FTP kann aber auch genutzt werden, um über embedded FTP-Clients wie zum Beispiel den W&T Com-Server serielle Daten von Endgeräten in eine Datei auf dem Server zu speichern.

TFTP - Trivial File Transfer Protocol

Neben FTP ist TFTP ein weiterer Dienst, um übers Netzwerk auf die Dateien eines entfernten Rechners zugreifen zu können.

TFTP ist allerdings sowohl vom Funktionsumfang, als auch von der Größe des Programmcodes deutlich „schlanker“ als FTP.

Ein TFTP-Client ist nicht unbedingt Bestandteil des Betriebssystems und z.B. im Windows-Umfeld nur in Windows NT und Windows 2000 implementiert.

TFTP-Server kommen im Officebereich selten zum Einsatz.

Besonders geeignet ist TFTP für den Einsatz in Embedded Systemen, in denen nur ein begrenzter Speicherplatz für Betriebssystemkomponenten zur Verfügung steht. TFTP bietet hier bei minimalem Programmcod ein hohes Maß an Effizienz.

In Com-Servern, Printerservern und Miniterminals wird beispielsweise TFTP genutzt, um Konfigurations- und Firmware-Dateien zu übertragen.

TFTP stellt nur zwei Dateioperationen zur Verfügung:

| | TFTP Befehl |
|-------------------------------------|-------------|
| Speicher von Dateien auf dem Server | PUT |
| Laden von Dateien vom Server | GET |

Wie auch FTP unterscheidet TFTP zwischen der Übertragung von Text- und Binär-Dateien. Sollen Binär-Dateien übertragen werden, wird dies durch den zusätzlichen Parameter „-i“ angegeben.

Hier als kurzes Beispiel: Die binäre Datei „test.txt“ wird von einem Windows NT Rechner auf den Server wlinux gespeichert.

```

Eingabeaufforderung
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.
C:\>tftp -i wlinux put test.txt
250kb erfolgreich übertragen
C:\>

```

Auf eine Authentifizierung, also ein Login mit Passwortabfrage wie bei FTP, wird verzichtet.

Eine Möglichkeit, dennoch unerwünschte Zugriffe auszuschließen, möchten wir am Beispiel des Com-Servers zeigen. Um einem Com-Server via TFTP eine neue Firmware einzuspielen, muss zunächst der TFTP Zugang über Telnet freigeschaltet werden.

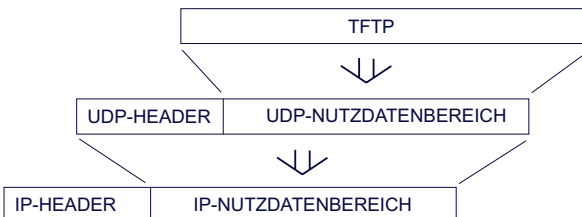
```

Telnet - 172.16.232.47
Verbinden Bearbeiten Terminal ?
*****
*          MINI Com-Server          *
*****
*** Port:- / Menu Level:2 ***
Flash Update
  1. Net Update (TFTP)
  2. Serial Update (Port 0)
Press <No.+ ENTER> (q=quit): 1
Flash Update ?(Y):

```

Darüber hinaus wird geprüft, ob es sich bei den empfangenen Daten tatsächlich um Com-Server Firmware handelt.

Im Gegensatz zu FTP verwendet TFTP als Basis Protokoll UDP, wobei der Port 69 genutzt wird.

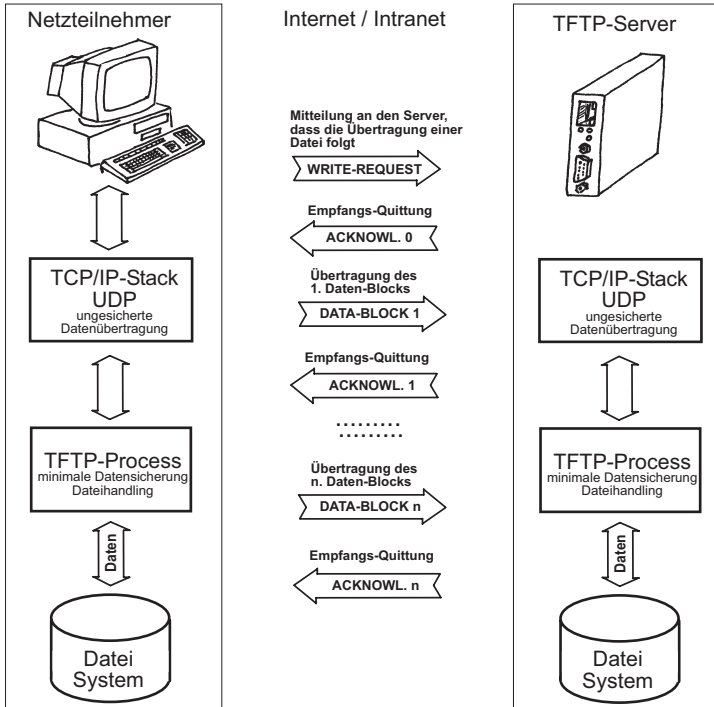


Zur Erinnerung:

UDP arbeitet verbindungslos. Man spricht bei UDP-Paketen auch von Datagrammen, wobei jedes Paket als eigenständige Datensendung behandelt wird. Auf UDP-Ebene werden empfangene Pakete nicht quittiert. Der Sender erhält keine Rückmeldung, ob ein gesendetes Paket wirklich beim Empfänger angekommen ist. UDP-Pakete bekommen keine Sequenz-Nummer. Ein Empfänger, der mehrere UDP-Pakete erhält, hat keine Möglichkeit festzustellen, ob die Pakete in der richtigen Reihenfolge empfangen wurden.

Aus diesem Grund übernimmt TFTP die Sicherung der übertragenen Daten selbst.

Die Übertragung von Dateien geschieht in Blöcken von je 512 Bytes, wobei die Blöcke mit einer laufenden Nummer versehen werden. Jeder empfangene Block wird von der Gegenseite quittiert. Erst nach Empfang der Quittung, wird der nächste Block gesendet.



TFTP erkennt, ob die empfangenen Datenblöcke in Ordnung sind, eine Fehlerkorrektur gibt es aber nicht. Geht bei der Übertragung etwas schief, stimmt etwa die Paketlänge nicht oder ein komplettes Paket geht verloren, wird die Übertragung abgebrochen. In diesem Fall kann der Anwender oder eine intelligente Anwendungssoftware den Vorgang erneut starten.

SNMP - Simple Network Management Protocol

Bei Drucklegung dieser Auflage war dieses Kapitel leider noch nicht fertiggestellt.

Ergänzungen zu diesem Buch finden Sie im Internet auf allen Com-Server Datenblatt Seiten als PDF-File.

Hier liegt auch immer die aktuelle Auflage zum Download bereit.

Besuchen Sie uns unter <http://www.wut.de>

Modbus-TCP

Bei Drucklegung dieser Auflage war dieses Kapitel leider noch nicht fertiggestellt.

Ergänzungen zu diesem Buch finden Sie im Internet auf allen Com-Server Datenblatt Seiten als PDF-File.

Hier liegt auch immer die aktuelle Auflage zum Download bereit.

Besuchen Sie uns unter <http://www.wut.de>

Socket-Programmierung

Die, in den vorherigen Kapiteln gezeigten, Standard-Internet-Protokolle und Dienste bieten bereits Lösungsmöglichkeiten für diverse Anwendungen.

Oft werden aber auch speziell auf den Anwendungsfall zugeschnittene Softwarelösungen benötigt. Das kann gleichermaßen spezielle Bedien- und Eingabeoberflächen auf Anwenderenebene, als auch technische Einbindung in Endgeräte und bestehende Programme betreffen.

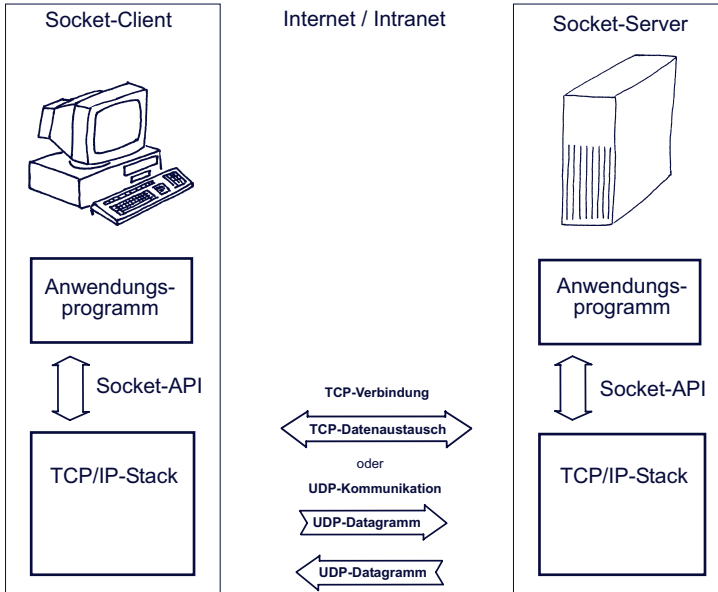
Zur Erinnerung: Den Teil eines Betriebssystems, der für die TCP/IP-Kommunikation zuständig ist, bezeichnet man als TCP/IP-Stack. Der TCP/IP-Treiber tauscht mit dem Anwendungsprogramm IP-Adressen und Ports, sowie zu übermittelnde Nutzdaten aus und stellt daraus IP-Pakete zusammen. Diese IP-Pakete werden vom TCP/IP-Stack zum physikalischen Versand an den Netzwerkkartentreiber weitergegeben.

Die eindeutige Zuordnung einer Verbindung entsteht aus dem Zusammenwirken von IP-Adresse und Port-Nummer. Man spricht bei dieser Zuordnung von einem Socket.

Wer eigene Anwendungen entwickeln möchte, die eine Kommunikation via TCP/IP unterstützen, hat mit der Socket-Programmierung nahezu uneingeschränkte Möglichkeiten.

Alle modernen Betriebssysteme verfügen heute über ein Socket-Application Interface.

Das Socket-API ist eine definierte Software-Schnittstelle, die je nach Programmiersprache und Betriebssystem, über DLL-Dateien oder Controls, Zugriff auf den TCP/IP-Stack erlaubt.



Eine besonders einfache Plattform für das Erstellen eigener Anwendungen bieten die Hochsprachen Visual Basic und Delphi, die hier mit kurzen Programmbeispielen vorgestellt werden.

Selbstverständlich bieten auch Programmiersprachen wie C++ und Java hervorragende Voraussetzungen für die Socket-Programmierung. Beispiele und Erklärungen hierzu finden Sie unter <http://www.wut.de>.

TCP-Client, TCP-Server oder UDP-Peer?

Unabhängig von der gewählten Entwicklungsumgebung, sollte anhand der Aufgabenstellung und der beteiligten Komponenten zunächst entschieden werden, welchen Teil der Kommunikation das zu erstellende Programm einnehmen soll.

TCP

Anwendungen, bei denen größere Datenmengen ausgetauscht werden, sollten auf Basis von TCP erstellt werden. TCP hat hier

den Vorteil einer festen Verbindung, bei der die Sicherung der Daten vom TCP/IP-Stack übernommen wird.

TCP-Client

Soll das eigene Programm bestimmen, wann und mit wem Verbindung aufgenommen wird, ist es sinnvoll eine Client-Anwendung zu programmieren.

Der TCP/IP-Stack benötigt vom Anwendungsprogramm die IP-Adresse bzw. den Hostnamen des Servers und die Port-Nummer, auf der der Server eine Verbindung entgegennimmt.

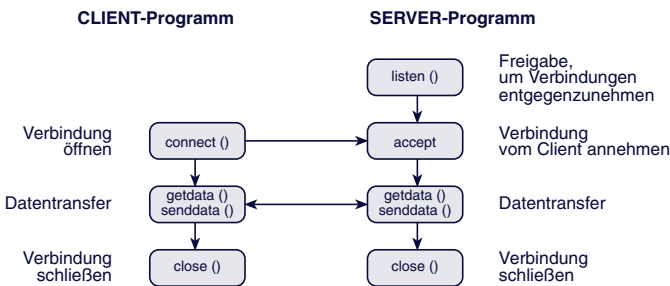
Der lokale Port auf dem die Client-Anwendung Daten vom Server empfängt, wird in Client-Anwendungen vom TCP/IP-Stack willkürlich vergeben.

TCP-Server

Wenn die eigene Applikation dagegen einer oder mehreren anderen Anwendungen Daten und Dienste zur Verfügung stellen soll, wird ein TCP-Servers programmiert.

Der TCP/IP-Stack benötigt vom Anwendungsprogramm die Information, auf welchem lokalen Port Verbindungswünsche einer Client-Anwendung entgegengenommen werden sollen.

Liegt ein Verbindungswunsch vor, informiert der Stack das Programm. Akzeptiert das Anwendungsprogramm die Verbindung, können Daten gesendet und empfangen werden.



UDP

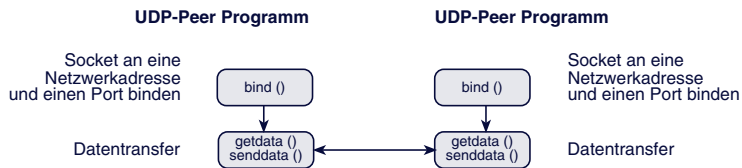
Bei Netzerwerkanwendungen mit wechselnden Partnern oder nur kurzen Datensendungen ist UDP als verbindungsloses Protokoll vorzuziehen.

Auf UDP-Ebene sind beide Kommunikationspartner gleichberechtigt. Man unterscheidet nicht zwischen Client und Server. Da keine Zeit für Verbindungsauf- und Abbau verloren geht, können bei kleineren Datenmengen deutlich schnellere Zugriffszeiten erreicht werden.

Es sollte aber beachtet werden, dass die Sicherung der Daten in der eigenen Applikation realisiert werden muss.

Der TCP/IP-Stack benötigt vom Anwendungsprogramm die IP-Adresse bzw. den Hostnamen und die Port-Nr. des Kommunikationspartners sowie die eigene Port-Nummer.

Durch die Zuordnung dieser Parameter entsteht ein Socket, über das Daten gesendet und empfangen werden können.




Socket-Programmierung in Visual Basic

Die vorgestellten Beispiele wurden in Visual Basic 5 erstellt, das auch nach Markteinführung von VB6 z.Zt. noch die größere Verbreitung genießt.

Alle, die über Grundkenntnisse in VB-Programmierung verfügen, sollten den Programmbeispielen leicht folgen können.

Um in VB TCP/IP basierende Anwendungen erstellen zu können, muss zunächst das Winsock-Control in die Komponentenliste aufgenommen werden.

- Mit der rechten Maustaste in die Komponentenleiste klicken
- Menüpunkt „Komponenten“ mit linker Maustaste auswählen
- In der Liste der Steuerelemente das Häkchen für „Microsoft Winsock Control“ setzen

Die Komponentenleiste ist nun um das Winsock-Steuer-element bereichert. 

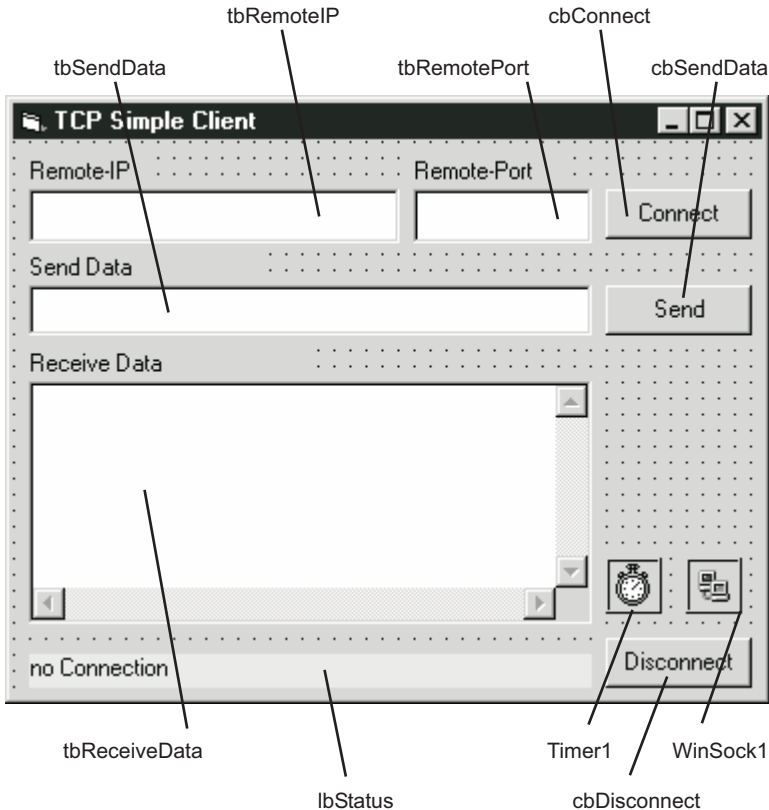
Ein TCP-Client in VB

Als erstes wollen wir einen TCP-Client erstellen, der folgende Aufgaben übernimmt:

(das komplette Beispiel, steht unter <http://www.wut.de> zum Download zur Verfügung)

- Aufbau der TCP-Verbindung
- Senden und Empfangen von Textdaten
- Schließen der TCP-Verbindung
- Anzeigen des Verbindungsstatus
- Erkennen von Fehlern

Hierzu wird ein Formular mit folgenden Elementen aufgebaut:



Alle Variablen und Elementnamen sollten sich durch die gewählte Namensgebung selbst erklären.

Für Elemente vom Typ Textbox wurden mit „tb“ beginnende Namen gewählt, Command-Buttons beginnen dagegen mit „cb“.

Der gezeigte VB-Quelltext kommt deshalb auch weitestgehend ohne Kommentare aus:

Die folgende Prozedur setzt nach Klick auf den Connect-Button, die vom Anwender eingegebenen Adressierungsparameter und baut, mit Hilfe der Connect-Methode des Winsock-Controls, die TCP-Verbindung auf.

```
Private Sub cbConnect_Click()
```

```
If (tbRemotePort.Text <> "") And (tbRemoteIP.Text <> "") Then
    Winsock1.RemotePort = tbRemotePort.Text
    Winsock1.RemoteHost = tbRemoteIP.Text
    Winsock1.Connect
End If
End Sub
```

Prozedur zum Trennen der TCP-Verbindung mit Hilfe der Close-Methode.

```
Private Sub cbDisconnect_Click()
    Winsock1.Close
    cbConnect.Enabled = True
    cbConnect.SetFocus
End Sub
```

Durch Klicken auf das Send-Button wird der vom Anwender eingegebene Text über die bestehende TCP-Verbindung versandt. Hierzu wird die Senddata Methode benutzt.

```
Private Sub cbSendData_Click()
    Winsock1.SendData (tbSendData.Text)
    tbSendData.Text = ""
End Sub
```

Timerroutine überwacht den aktuellen Verbindungsstatus über die State Eigenschaft des Winsock-Controls. Ein sinnvoller Intervall für den Timer ist 500ms.

```
Private Sub Timer1_Timer()
    Select Case Winsock1.State
        Case ckClosed
            lbStatus.Caption = "no Connection"
        Case sckResolvingHost
            lbStatus.Caption = "waiting for DNS"
        Case sckHostResolved
            lbStatus.Caption = "get IP from DNS"
        Case sckConnecting
            lbStatus.Caption = "connecting"
        Case sckConnected
            lbStatus.Caption = "Connection to " + Winsock1.RemoteHost
        Case sckClosing
```

```
        lbStatus.Caption = "closing Connection"
    Case sockError
        lbStatus.Caption = "Connection Error"
        Winsock1.Close
    End Select
    If Winsock1.State <> sockConnected Then
        cbSendData.Enabled = False
        cbDisconnect.Enabled = False
        cbConnect.Enabled = True
    Else
        cbSendData.Enabled = True
        cbDisconnect.Enabled = True
        cbConnect.Enabled = False
    End If
End Sub
```

Diese Prozedur wird automatisch aufgerufen, wenn eine Verbindung von der Gegenseite beendet wird und kann z.B. genutzt werden, um über die Close-Methode auch die eigene Verbindungsverwaltung zurückzusetzen.

```
Private Sub Winsock1_Close()
    Winsock1.Close
    cbConnect.Enabled = True
    cbDisconnect.Enabled = False
    cbSendData.Enabled = False
    lbStatus.Caption = "no Connection"
    cbConnect.SetFocus
End Sub
```

Diese Prozedur wird automatisch bei erfolgreichem Verbindungsaufbau aufgerufen.

```
Private Sub Winsock1_Connect()
    cbDisconnect.Enabled = True
    cbConnect.Enabled = False
    tbReceiveData.Text = ""
    lbStatus.Caption = "connected to " + Winsock1.RemoteHost
End Sub
```

Bei Datenempfang wird automatisch diese Prozedur durchlaufen.

Eingehende Daten werden mit der Getdata-Methode entgegengenommen und im „Receive Data“ Fenster angezeigt.

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
    Dim ReceiveData As String
    Winsock1.GetData ReceiveData
    tbReceiveData.Text = tbReceiveData.Text + ReceiveData
End Sub
```

Prozedur zur Behandlung von Verbindungsfehlern

```
Private Sub Winsock1_Error(ByVal Number As Integer, _
    Description As String, ByVal Scode As Long, _
    ByVal Source As String, ByVal HelpFile As String, _
    ByVal HelpContext As Long, CancelDisplay As Boolean)
    Winsock1.Close
    dummy = MsgBox("Connection Error", vbOKOnly, "TCP simple Client")
End Sub
```

So hat man mit weniger als 2 Seiten Quelltext bereits einen TCP-Client mit Statusanzeige und Fehlerbehandlung programmiert.

Als Gegenstück wird natürlich ein Server benötigt, der den Verbindungswunsch des Clients annimmt. Das kann ein vorhandenes Gerät, wie z.B. ein W&T Com-Server oder eine weitere VB-Server-Applikation sein.

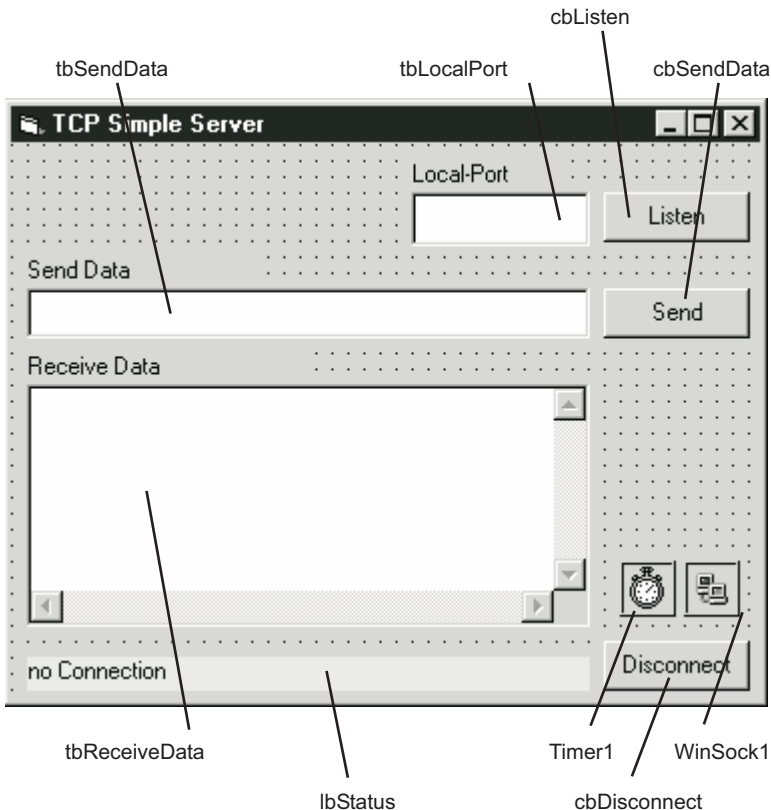
Wie ein TCP-Server aufgebaut werden kann, soll im folgenden Beispiel gezeigt werden.

Ein TCP-Server in VB

Die Server-Applikation übernimmt folgende Aufgaben:

- Auf dem Netz „Horchen“, ob es auf dem unterstützten Port einen Verbindungswunsch gibt
- Gewünschte Verbindung entgegennehmen
- Senden und Empfangen von Textdaten
- Anzeigen des Verbindungsstatus
- Anzeigen von Verbindungsfehlern
- Schließen der Verbindung von der Server-Seite

Hierzu wird ein Formular mit folgenden Elementen aufgebaut:



Alle Variablen und Elementnamen sollten sich auch hier durch die gewählte Namensgebung selbst erklären.

Die folgenden VB-Prozeduren werden für den Server benötigt:

Diese Prozedur wird für eine Server-Anwendung nicht unbedingt benötigt, und steht nur am Anfang des Quelltextes, weil VB die Prozeduren alphabetisch sortiert anzeigt. Sie erlaubt es, mit der Close-Methode eine bestehende Verbindung zu schließen. Mit der Listen-Methode beginnt das Winsock-Control erneut auf etwaige Verbindungswünsche auf den gewählten Port zu horchen.

```
Private Sub cbDisconnect_Click()  
    Winsock1.Close  
    Winsock1.Listen  
End Sub
```

Mit Aufruf der Listen-Methode, beginnt das Winsock-Control auf etwaige Verbindungswünsche auf den gewählten Port zu horchen.

```
Private Sub cbListen_Click()  
If tbLocalPort.Text <> "" Then  
    Winsock1.LocalPort = tbLocalPort.Text  
    Winsock1.Listen  
    cbListen.Enabled = False  
End If  
End Sub
```

Durch Klicken auf das Send-Button wird der vom Anwender eingegebene Text über die bestehende TCP-Verbindung versandt. Hierzu wird die Senddata Methode benutzt.

```
Private Sub cbSendData_Click()  
    Winsock1.SendData (tbSendData.Text)  
    tbSendData.Text = ""  
End Sub
```

Die Timerroutine überwacht den aktuellen Verbindungsstatus über die State-Eigenschaft des Winsock-Controls. Ein sinnvolles Intervall für den Timer ist 500ms.

```
Private Sub Timer1_Timer()  
    Select Case Winsock1.State  
        Case ckClosed  
            lbStatus.Caption = "no Connection"  
        Case sckListening  
            lbStatus.Caption = "listening for connection"  
        Case sckConnectionPending  
            lbStatus.Caption = "Connection Pending"  
        Case sckConnecting  
            lbStatus.Caption = "connecting"  
        Case sckConnected  
            lbStatus.Caption = „Connection to „ + Winsock1.RemoteHostIP  
        Case sckError  
            lbStatus.Caption = "Connection Error"  
            Winsock1.Close  
    End Select  
    If Winsock1.State <> sckConnected Then  
        cbSendData.Enabled = False  
        cbDisconnect.Enabled = False  
    Else  
        cbSendData.Enabled = True  
        cbDisconnect.Enabled = True  
    End If  
End Sub
```

Diese Prozedur wird automatisch aufgerufen, wenn eine Verbindung von der Client-Seite beendet wird. Über die Close-Methode wird die eigene Verbindungsverwaltung zurückgesetzt. Mit Aufruf der Listen-Methode, beginnt das Winsock-Control erneut, auf etwaige Verbindungswünsche auf den gewählten Port zu horchen.

```
Private Sub Winsock1_Close()  
    Winsock1.Close  
    Winsock1.Listen  
End Sub
```

Erkennt das Winsock-Steuerelement den Verbindungswunsch eines Client, wird automatisch diese Prozedur ausgeführt. Mit der Accept-Methode wird die Verbindung entgegengenommen

```
Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)  
    If Winsock1.State <> sckclose Then Winsock1.Close  
    Winsock1.Accept requestID
```

```
End Sub
```

Bei Datenempfang wird automatisch diese Prozedur durchlaufen. Eingehende Daten werden mit der Getdata-Methode entgegengenommen und im „Receive Data“ Fenster angezeigt.

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
    Dim ReceiveData As String
    Winsock1.GetData ReceiveData
    tbReceiveData.Text = tbReceiveData.Text + ReceiveData
End Sub
```

Prozedur zur Behandlung von Verbindungsfehlern

```
Private Sub Winsock1_Error(ByVal Number As Integer, Description As String, ByVal
Scode As Long, ByVal Source As String, ByVal HelpFile As String, ByVal HelpContext
As Long, CancelDisplay As Boolean)
    Winsock1.Close
    Winsock1.LocalPort = 0
    dummy = MsgBox("Connection Error", vbOKOnly, "TCP simple Server")
End Sub
```

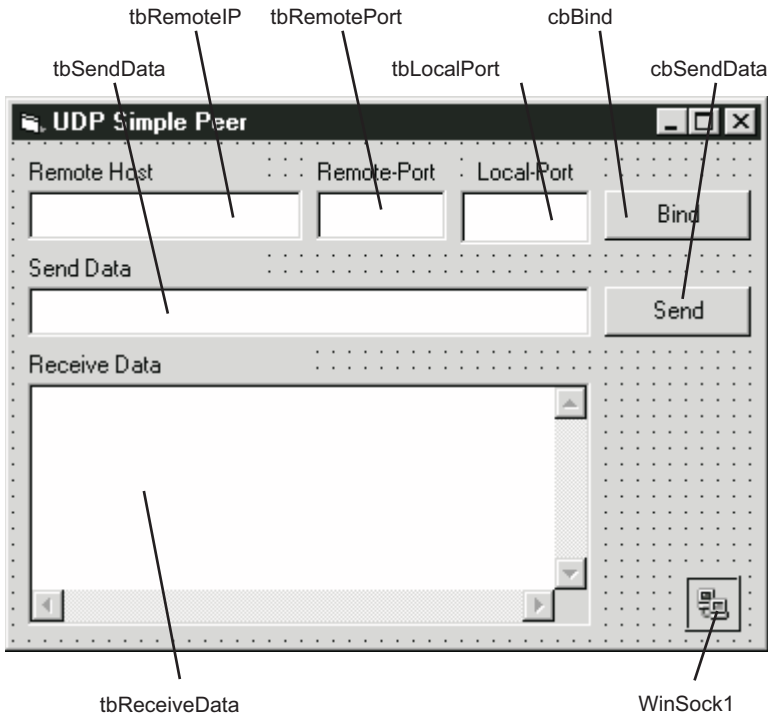
Auf den so programmierten Server kann man mit dem vorgestellten Client-Programm zugreifen. Aber auch beliebige andere Client-Anwendungen, z.B. der W&T Com-Server im Client-Mode, können bei Wahl des entsprechenden Ports Verbindung mit dem Server aufnehmen.

Ein einfacher UDP-Peer in VB

Die UDP-Applikation übernimmt folgende Aufgaben:

- IP-Adresse und Ports zu einem Socket binden
- Senden und Empfangen von Textdaten

Hierzu wird ein Formular mit folgenden Elementen aufgebaut:



Alle Variablen und Elementnamen sollten sich auch hier durch die gewählte Namensgebung selbst erklären.

Der folgende Quelltext kommt deshalb auch weitestgehend ohne Kommentare aus:

Mit der Bind-Methode werden Ip-Adresse und Ports in einem Socket gebunden

```
Private Sub cbBind_Click()  
    Winsock1.Protocol = sckUDPProtocol  
    Winsock1.RemotePort = tbRemotePort.Text  
    Winsock1.RemoteHost = tbRemoteIP.Text  
    Winsock1.Bind tbLocalPort.Text  
    cbBind.Enabled = False  
    cbSendData.Enabled = True  
End Sub
```

Durch Klicken auf den Send-Button wird der vom Anwender eingegebene Text als UDP-Datagramm versandt. Hierzu wird die Senddata Methode benutzt. Bedingung hierfür ist, dass über die Bind-Methode ein Socket gebunden wurde.

```
Private Sub cbSendData_Click()  
    Winsock1.SendData (tbSendData.Text)  
    tbSendData.Text = ""  
End Sub
```

Bei Datenempfang wird automatisch diese Prozedur durchlaufen. Eingehende Daten werden mit der Getdata-Methode entgegengenommen und im „Receive Data“ Fenster angezeigt.

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)  
    Dim ReceiveData As String  
    Winsock1.GetData ReceiveData  
    tbReceiveData.Text = tbReceiveData.Text + ReceiveData  
End Sub
```

Um mit dem UDP-Peer Datenkommunikation zu betreiben, kann der gleiche Peer auf einem zweiten PC gestartet werden. Ebenso ist es aber auch möglich, über den UDP-Peer mit einem W&T Com-Server zu kommunizieren, der als UDP-Client konfiguriert ist.

Der hier gezeigte UDP-Peer verzichtet auf jede Form von Datensicherheit. Das bedeutet, werden Daten an eine nicht existente IP-Adresse geschickt oder das adressierte Endgerät ist nicht betriebsbereit, laufen die Daten einfach ins Leere, ohne dass der Anwender etwas merkt.

Socket-Programmierung in Delphi

Die hier vorgestellten Beispiele wurden in der Delphi 5 Standardversion erstellt.

Alle, die über Grundkenntnisse in Delphi-Programmierung verfügen, sollten den Programmbeispielen leicht folgen können.

Delphi 5 stellt im Register „Internet“ Standard-Steuerelemente zur Socket-Programmierung zur Verfügung.

Im Gegensatz zu Visual Basic, wo nur ein Steuerelement durch unterschiedliche Parametrierung dem gewünschten Einsatz angepasst werden kann, bietet Delphi zwei spezifische Steuerelemente an:

Steuerelement für TCP-Client (ClientSocket)



Steuerelement für TCP-Server (ServerSocket)



Ein Steuerelement für UDP-Anwendungen ist in der Standardversion von Delphi 5 leider nicht vorhanden

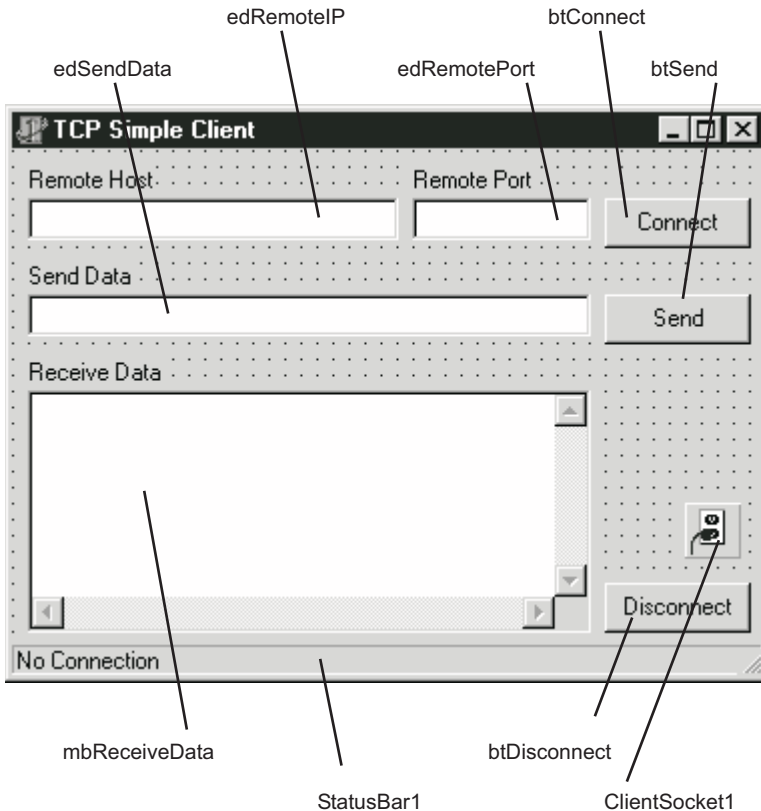
Ein TCP-Client in Delphi

Als erstes wollen wir einen TCP-Client erstellen, der folgende Aufgaben übernimmt:

(das komplette Beispiel steht unter <http://www.wut.de> zum Download zur Verfügung)

- Aufbau der TCP-Verbindung
- Senden und Empfangen von Textdaten
- Schließen der TCP-Verbindung
- Anzeigen des Verbindungsstatus
- Erkennen von Fehlern

Hierzu wird ein Formular mit folgenden Elementen aufgebaut:



Alle Variablen und Elementnamen sollten sich durch die gewählte Namensgebung selbst erklären.

Für Elemente vom Typ Edit wurden mit „ed“ beginnende Namen gewählt, Buttons beginnen dagegen mit „bt“ und Memoboxen mit „mb“

Der gezeigte Quelltext kommt deshalb auch weitestgehend ohne Kommentare aus:

Der erste Teil des Quelltextes wird von Delphi beim Entwerfen des Formulars selbst erstellt, und dient der Deklaration aller beteiligten Elemente.


```
unit TCP_Client;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ScktComp, StdCtrls, ComCtrls;

type
  TTCPCClient = class(TForm)
    edRemoteIP: TEdit;
    btConnect: TButton;
    edRemotePort: TEdit;
    edSendData: TEdit;
    btSend: TButton;
    mbReceiveData: TMemo;
    btDisconnect: TButton;
    StatusBar1: TStatusBar;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    ClientSocket1: TClientSocket;
    procedure btConnectClick(Sender: TObject);
    procedure btSendClick(Sender: TObject);
    procedure OnConnect(Sender: TObject; Socket: TCustomWinSocket);
    procedure btDisconnectClick(Sender: TObject);
    procedure OnDisconnect(Sender: TObject; Socket: TCustomWinSocket);
    procedure OnRead(Sender: TObject; Socket: TCustomWinSocket);
    procedure OnError(Sender: TObject; Socket: TCustomWinSocket);
    ErrorEvent: TErrorEvent; var ErrorCode: Integer);
  private
    { Private-Deklarationen }
  public
    { Public-Deklarationen }
  end;

var
  TCPClient: TTCPCClient;

implementation
```

```
{ $R *.DEM }
```

Hier beginnt das eigentliche Programm

Die folgende Prozedur setzt nach Klick auf den Connect-Button, die vom Anwender eingegebenen Adressierungsparameter und baut durch Aktivieren des Winsock-Controls die TCP-Verbindung auf.

```
procedure TTCPCClient.btConnectClick(Sender: TObject);
begin
    ClientSocket1.Host := edRemoteIP.Text;
    ClientSocket1.Port := strtoint(edRemotePort.Text);
    ClientSocket1.Active := True;
end;
```

Diese Prozedur wird automatisch bei erfolgreichem Verbindungsaufbau aufgerufen.

```
procedure TTCPCClient.OnConnect(Sender: TObject; Socket: TCustomWinSocket);
begin
    btSend.Enabled := True;
    btDisconnect.Enabled := True;
    btConnect.Enabled := False;
    mbReceiveData.Clear;
    StatusBar1.SimpleText := 'Connected to ' + ClientSocket1.Host;
end;
```

Diese Prozedur wird automatisch bei Verbindungsabbau aufgerufen.

```
procedure TTCPCClient.OnDisconnect(Sender: TObject;
    Socket: TCustomWinSocket);
begin
    btSend.Enabled := False;
    btDisconnect.Enabled := False;
    btConnect.Enabled := True;
    StatusBar1.SimpleText := 'No Connection';
```

```
end;
```

Prozedur zur automatischen Fehlerbehandlung.

```
procedure TTCPCClient.OnError(Sender: TObject; Socket: TCustomWinSocket;  
  ErrorEvent: TErrorEvent; var ErrorCode: Integer);  
begin  
  ShowMessage ('Connection Error');  
  ClientSocket1.Active := False;  
  btSend.Enabled := False;  
  btDisconnect.Enabled := False;  
  btConnect.Enabled := True;  
  StatusBar1.SimpleText := 'No Connection';  
end;
```

Durch klicken auf den Send-Button wird der vom Anwender eingegebene Text über die bestehende TCP-Verbindung versandt.

```
procedure TTCPCClient.btSendClick(Sender: TObject);  
begin  
  ClientSocket1.Socket.SendText (edSendData.Text);  
  edSendData.Text := '';  
end;
```

*Bei Datenempfang wird automatisch diese Prozedur durchlaufen.
Eingehende Daten werden entgegengenommen und im „Receive Data“ Fenster angezeigt.*

```
procedure TTCPCClient.OnRead(Sender: TObject; Socket: TCustomWinSocket);  
begin  
  mbReceiveData.Text := mbReceiveData.Text + ClientSocket1.Socket.ReceiveText;  
end;
```

Prozedur zum Trennen der TCP-Verbindung durch Deaktivierung des ClientSocket Steuerelements.

```
procedure TTCPClient.btDisconnectClick(Sender: TObject);  
begin  
    ClientSocket1.Active := False;  
end;  
  
end.
```

So hat man auch in Delphi mit weniger als 2 Seiten Quelltext bereits einen TCP-Client mit Statusanzeige und Fehlerbehandlung programmiert.

Als Gegenstück wird natürlich ein Server benötigt, der den Verbindungswunsch des Clients annimmt. Das kann ein vorhandenes Gerät wie, z.B. ein W&T Com-Server, oder eine weitere Delphi-Server Applikation sein.

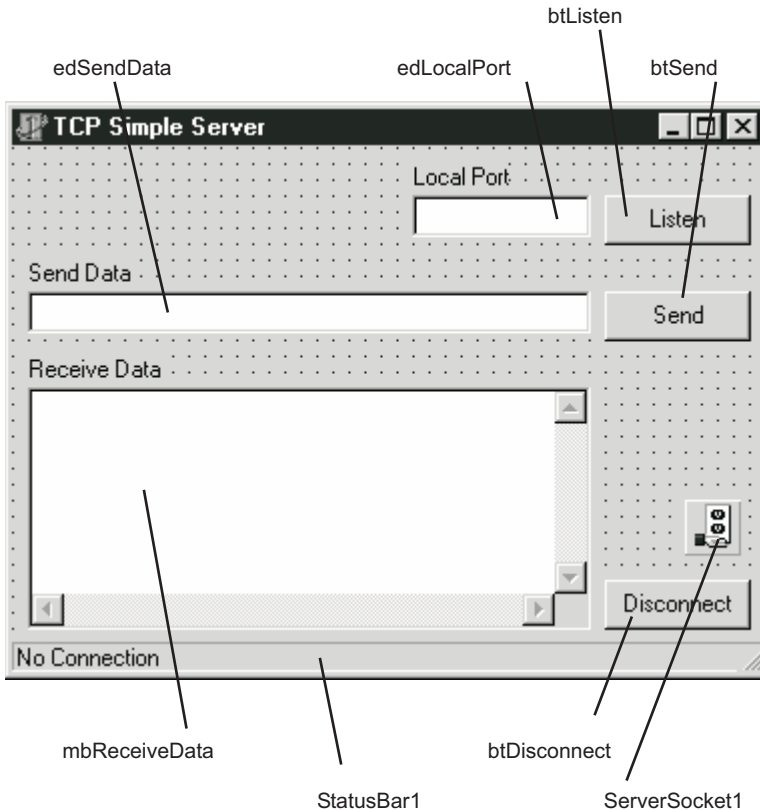
Wie ein TCP-Server in Delphi aufgebaut werden kann, soll im folgenden Beispiel gezeigt werden.

Ein TCP-Server in Delphi

Die Server-Applikation übernimmt folgende Aufgaben:

- Auf dem Netz „Horchen“, ob es auf dem unterstützten Port einen Verbindungswunsch gibt
- Gewünschte Verbindung entgegennehmen
- Senden und Empfangen von Textdaten
- Anzeigen des Verbindungsstatus
- Anzeigen von Verbindungsfehlern
- Schließen der Verbindung von der Server-Seite (gehört nicht zu den typischen Aufgaben eines Servers, ist mit dem Programm aber auch möglich)

Hierzu wird ein Formular mit folgenden Elementen aufgebaut:



Alle Variablen und Elementnamen sollten sich auch hier durch die gewählte Namensgebung selbst erklären.

Die folgenden Delphi-Prozeduren werden für den Server benötigt:

Wie schon bei der Client-Anwendung, wird der erste Teil des Quelltextes von Delphi beim Entwerfen des Formulars selbst erstellt, und dient der Deklaration aller beteiligten Elemente.

```
unit TCP_Server;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
```

```
ScktComp, StdCtrls, ComCtrls;
```

```
type
```

```
  TTCPServer = class(TForm)
    btListen: TButton;
    edLocalPort: TEdit;
    edSendData: TEdit;
    btSend: TButton;
    mbReceiveData: TMemo;
    btDisconnect: TButton;
    StatusBar1: TStatusBar;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    ServerSocket1: TServerSocket;
    procedure btListenClick(Sender: TObject);
    procedure btSendClick(Sender: TObject);
    procedure btDisconnectClick(Sender: TObject);
    procedure OnListen(Sender: TObject; Socket: TCustomWinSocket);
    procedure OnAccept(Sender: TObject; Socket: TCustomWinSocket);
    procedure OnClientRead(Sender: TObject; Socket: TCustomWinSocket);
    procedure OnClientDisconnect(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure OnClientError(Sender: TObject; Socket: TCustomWinSocket;
      ErrorEvent: TErrorEvent; var ErrorCode: Integer);
  private
    { Private-Deklarationen }
  public
    { Public-Deklarationen }
  end;
```

```
var
```

```
  TCPServer: TTCPServer;
```

```
implementation
```

```
{ $R *.DFM }
```

Hier beginnt das eigentliche Programm:

Durch Klick auf den Listen Button wird das ServerSocket Steuerelement geöffnet und beginnt auf etwaige Verbindungswünsche auf den gewählten Port zu horchen.

```
procedure TTCPServer.btListenClick(Sender: TObject);
begin
    If edLocalPort.Text <> '' Then
        begin
            ServerSocket1.Port := strtoint(edLocalPort.Text);
            ServerSocket1.Open;
        end
    Else ShowMessage ('No local port!');
end;
```

Diese Prozedur wird automatisch aufgerufen, wenn das ServerSocket Steuerelement geöffnet wurde und auf Verbindungswünsche wartet.

```
procedure TTCPServer.OnListen(Sender: TObject; Socket: TCustomWinSocket);
begin
    StatusBar1.SimpleText := 'Listening';
    btSend.Enabled := False;
    btDisconnect.Enabled := False;
    btListen.Enabled := False;
end;
```

Das Entgegennehmen von Verbindungen erledigt das ServerSocket Steuerelement automatisch im Hintergrund. Wird eine Verbindung entgegengenommen, durchläuft das Programm automatisch die folgende Prozedur.

```
procedure TTCPServer.OnAccept(Sender: TObject; Socket: TCustomWinSocket);
begin
    StatusBar1.SimpleText := 'Connected to ' + Socket.RemoteAddress;
    btSend.Enabled := True;
    btDisconnect.Enabled := True;
end;
```

Diese Prozedur wird automatisch bei Verbindungsabbau aufgerufen.

```
procedure TTCPServer.OnClientDisconnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  StatusBar1.SimpleText := 'Listening';
  ServerSocket1.Open;
  btSend.Enabled := False;
  btDisconnect.Enabled := False;
end;
```

Prozedur zur automatischen Fehlerbehandlung.

```
procedure TTCPServer.OnClientError(Sender: TObject;
  Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
  var ErrorCode: Integer);
begin
  ShowMessage ('Connection Error');
  ErrorCode := 0;
  ServerSocket1.Close;
  btSend.Enabled := False;
  btDisconnect.Enabled := False;
  btListen.Enabled := True;
  StatusBar1.SimpleText := 'No Connection';
end;
```

Durch Klicken auf das Send-Button wird der vom Anwender eingegebene Text über die bestehende TCP-Verbindung versandt.

```
procedure TTCPServer.btSendClick(Sender: TObject);
begin
  ServerSocket1.Socket.Connections[0].SendText(edSendData.Text);
  edSendData.Text := '';
end;
```


*Bei Datenempfang wird automatisch diese Prozedur durchlaufen.
Eingehende Daten werden entgegengenommen und im „Receive Data“ Fenster angezeigt.*

```
procedure TTCPServer.OnClientRead(Sender: TObject;  
    Socket: TCustomWinSocket);  
begin  
    mbReceiveData.Text := mbReceiveData.Text + Socket.ReceiveText;  
end;
```

Prozedur zum Trennen der TCP-Verbindung durch Schließen des ServerSocket Steuerelements.

```
procedure TTCPServer.btDisconnectClick(Sender: TObject);  
begin  
    ServerSocket1.Close;  
    btListen.Enabled := True;  
    btSend.Enabled := False;  
    btDisconnect.Enabled := False;  
    StatusBar1.SimpleText := 'No Connection';  
end;  
  
end.
```

Auf den so programmierten Server kann man mit dem vorgestellten Client-Programm zugreifen. Aber auch beliebige andere Client-Anwendungen, z.B. der W&T Com-Server im Client-Mode, können bei Wahl des entsprechenden Ports Verbindung mit dem Server aufnehmen.

Wer mit Delphi UDP-Anwendungen programmieren möchte, hat die Möglichkeit, Steuerelemente von Drittanbietern einzusetzen.

Ein Beispiel hierfür ist das Internet Steuerelement des Belgiers Francois Piette, das unter <http://users.swing.be/francois.piette/indexuk.htm> kostenfrei zum Download bereitliegt.

Die gezeigten Beispiele sind als Anregung gedacht und sollen zum Ausprobieren und Spielen mit der Datenübertragung via TCP/IP einladen. Die Quelltexte können leicht durch Modifikation an gewünschte Anwendungswünsche angepasst werden.

TCP/IP -Ethernet Einrichten

Alle aktuellen Betriebssysteme bieten heute die Möglichkeit TCP/IP als lokales Netzwerkprotokoll zu nutzen.

Die Bedingung hierfür ist, dass der PC über eine Ethernet-Netzwerkkarte verfügt.


Wie das TCP/IP Protokoll auf den gängigen Microsoft Windows Systemen eingerichtet und konfiguriert wird, soll auf den folgenden Seiten Schritt für Schritt beschrieben werden.

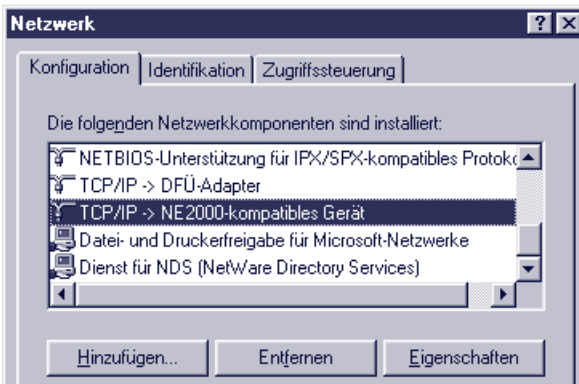
Wenn Ihr PC bereits in ein Ethernet-Netzwerk eingebunden ist, sollten Sie zunächst in Erfahrung bringen, ob in diesem Netzwerk bereits Anwendungen unter TCP/IP betrieben werden. Fragen Sie in diesem Fall Ihren Netzwerkadministrator, ob für Ihren PC schon eine IP-Adresse vorgesehen ist oder welche IP-Adressen Sie für Ihren PC verwenden können. Ferner müssen Sie wissen, welche Subnet-Mask, welches Gateway und welcher DNS-Server ggf. für das Netz gültig sind.

Bitte notieren Sie sich die verwendeten Werte:

| | | | | |
|-------------|--------|--------|--------|--------|
| IP-Adresse | _____. | _____. | _____. | _____. |
| Subnet-Mask | _____. | _____. | _____. | _____. |
| Gateway | _____. | _____. | _____. | _____. |
| DNS-Server | _____. | _____. | _____. | _____. |

TCP/IP unter Windows 9x installieren und konfigurieren

1. Klicken Sie auf *Start* und öffnen unter *Einstellungen* die *Systemsteuerung*.
2. Doppelklicken Sie auf das Netzwerk-Symbol .
3. Kontrollieren Sie, ob im Konfigurationsfenster *TCP/IP* -> „*Netzwerkkarte*“ aufgelistet ist.

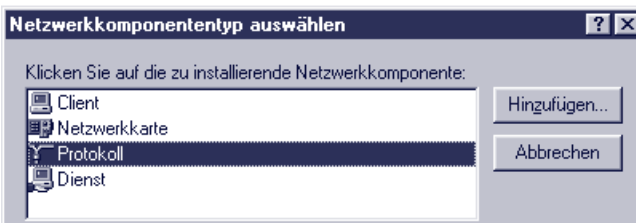


Wenn der Eintrag *TCP/IP* -> „*Netzwerkkarte*“ vorhanden ist, fahren Sie mit Punkt 5 fort.

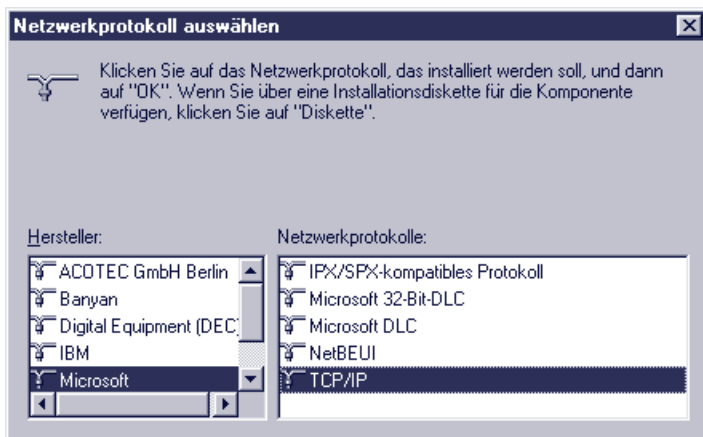


Der Eintrag *TCP/IP* -> *DFÜ-Adapter* reicht nicht aus, um *TCP/IP* unter *Ethernet* betreiben zu können!

4. Bei fehlendem Eintrag *TCP/IP* -> „*Netzwerkkarte*“ klicken Sie auf *Hinzufügen* und wählen im folgenden Fenster *Protokoll*.



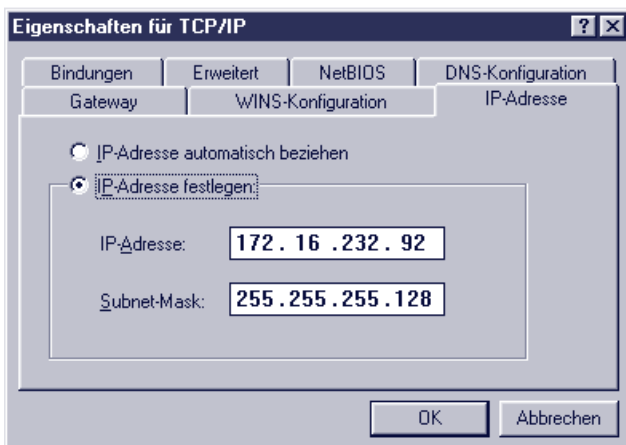
Klicken sie auf *Hinzufügen* und wählen Sie im folgenden Fenster als Hersteller *Microsoft* und als Netzwerkprotokoll *TCP/IP*.



Bestätigen Sie mit *OK*.

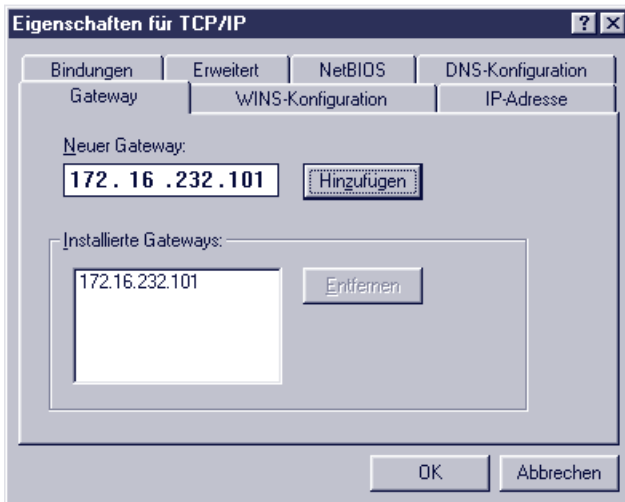
Zur Installation des Protokolls benötigen Sie die Installations-CD Ihrer Windows-Version.

5. Markieren Sie *TCP/IP->„Netzwerkkarte“* und wählen Sie *Eigenschaften*. Fragen Sie Ihren Netzwerkadministrator, ob die IP-Adresse über DHCP automatisch bezogen wird.



Wenn nicht, tragen Sie IP-Adresse und Subnet-Mask ein.

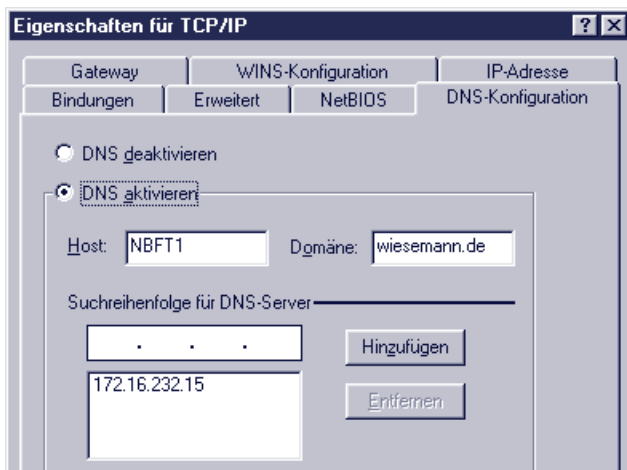
Wechseln Sie nun zum Register Gateway



Tragen Sie die IP-Adresse des Gateways im Feld *neuer Gateway* ein und klicken auf *Hinzufügen*. Nur wenn die eingetragene Gateway-Adresse im unteren Fenster erscheint, bleibt sie nach Bestätigung mit *OK* erhalten.

Arbeitet Ihr Netzwerk mit DNS-Unterstützung, sollte im Register DNS-Konfiguration auch die IP-Adresse des DNS-Servers eingetragen werden. Nur wenn die eingetragene DNS-Adresse im unteren Fenster erscheint, bleibt sie nach Bestätigung mit *OK* erhalten..


Ferner sollten Sie dort den Hostnamen des PC und die Domäne, in der er verwaltet wird, eintragen.

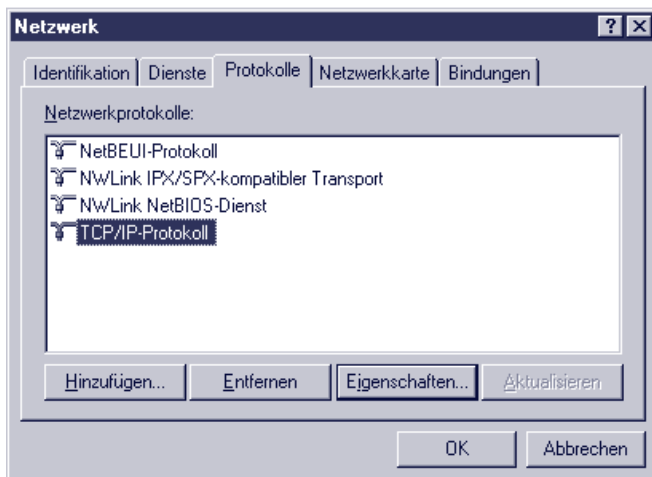


Bestätigen Sie mit *OK*.

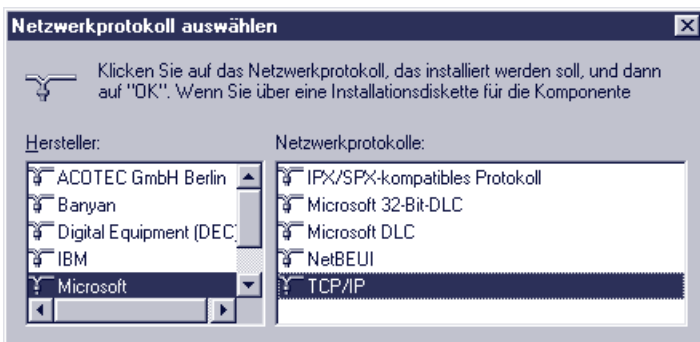
Damit ist die Installation von TCP/IP abgeschlossen, und Sie werden jetzt aufgefordert, Ihren PC neu zu starten.

TCP/IP unter Windows NT installieren u. konfigurieren

1. Klicken Sie auf *Start* und öffnen unter *Einstellungen* die *Systemsteuerung*.
2. Doppelklicken Sie auf das Icon .
3. Wenn im Register *Protokolle* *TCP/IP-Protokoll* aufgelistet ist können Sie an Punkt 5 fortfahren..



4. Bei fehlendem Eintrag *TCP/IP-Protokoll* klicken Sie auf *Hinzufügen* und wählen im folgenden Fenster *TCP/IP*.



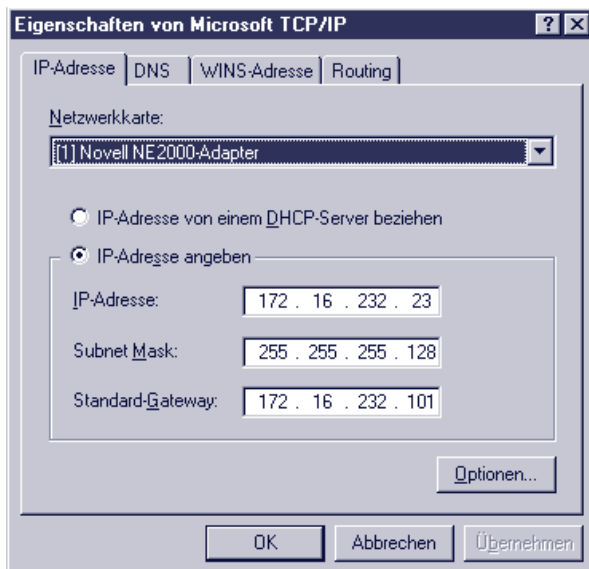
Sie benötigen nun die Windows-NT-Installations-CD. Bestätigen Sie mit *OK*.

- Bei neu hinzugefügter TCP/IP-Unterstützung klicken Sie *OK*, um die Eigenschaften zu konfigurieren. War TCP/IP bereits auf Ihrem PC installiert, markieren Sie den Eintrag *TCP/IP-Protokoll* und klicken Sie anschließend auf *Eigenschaften*.

Haben Sie die TCP/IP-Unterstützung neu installiert, erscheint nun folgende Meldung:

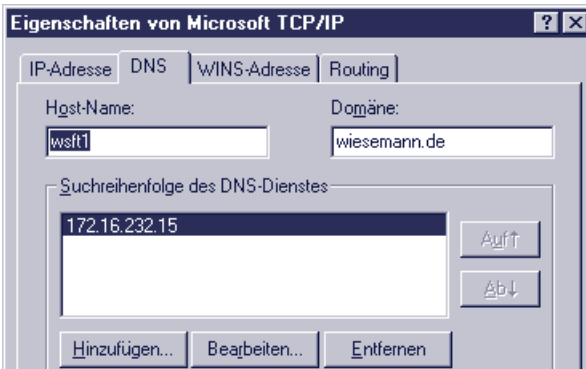


Fragen Sie Ihren Netzwerkadministrator, ob der DHCP-Dienst unterstützt wird. Ist das nicht der Fall, klicken Sie hier auf *Nein*.



Tragen Sie im folgenden Fenster IP-Adresse, Subnet-Mask und Gateway ein.

Arbeitet Ihr Netzwerk mit DNS-Unterstützung, sollte im Register *DNS-Konfiguration* auch die IP-Adresse des DNS-Servers eingetragen werden.



Ferner sollten Sie dort den Hostnamen des PC und die Domäne in der er verwaltet wird eintragen.

Bestätigen Sie mit *OK*.

Damit ist die Installation von TCP/IP abgeschlossen, und Sie werden jetzt aufgefordert, Ihren PC neu zu starten.

TCP/IP unter Win 2000 installieren und konfigurieren

1. Klicken Sie auf *Start* und öffnen unter *Einstellungen* die *Systemsteuerung*.
2. Doppelklicken Sie auf das Icon:



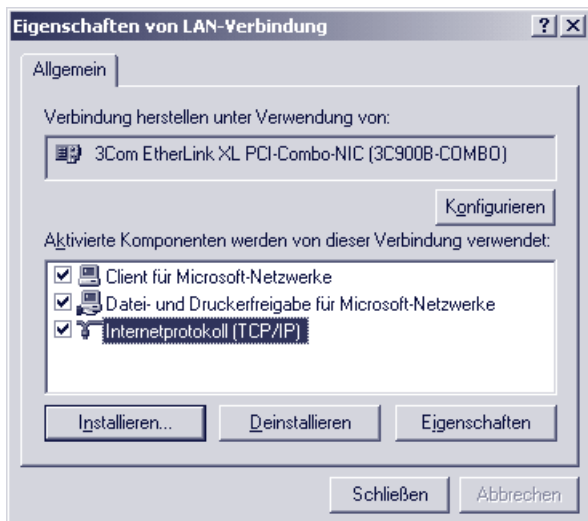
Netzwerk- und
DFÜ-Verbindungen

und im nächsten Fenster auf:



LAN-Verbindung

3. Kontrollieren Sie, ob *Internetprotokoll (TCP/IP)* aufgestellt ist.

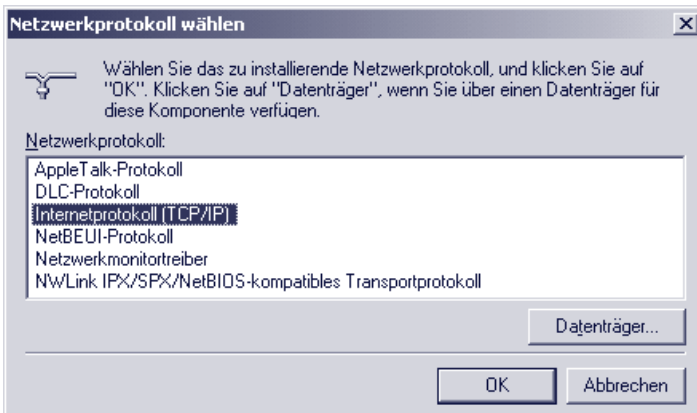


Wenn der Eintrag *Internetprotokoll (TCP/IP)* vorhanden ist, fahren Sie mit Punkt 5 fort.

4. Bei fehlendem Eintrag *Internetprotokoll (TCP/IP)* klicken Sie auf *Installieren* und wählen im folgenden Fenster *Protokoll* und *Hinzufügen*.



Wählen Sie *Internetprotokoll (TCP/IP)*.

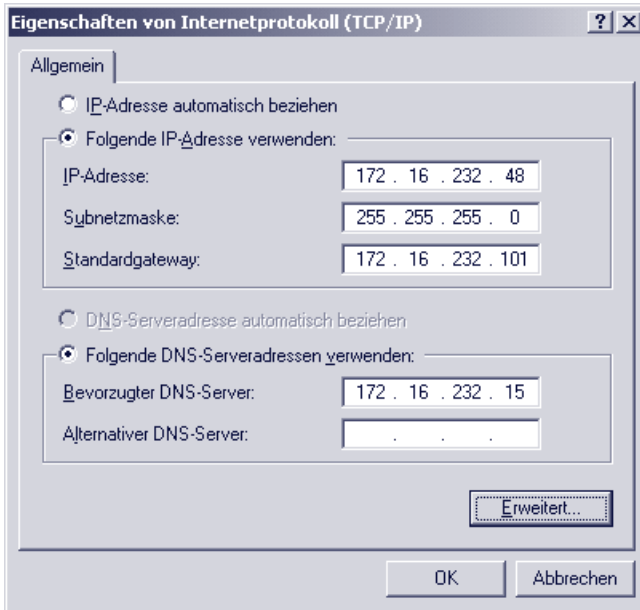


Sie benötigen nun die Windows 2000 Installations-CD. Nach Bestätigung mit *OK*, ist die Liste der Netzwerkprotokolle um den Eintrag *TCP/IP-Protokoll* erweitert.

5. Es erscheint nun wieder das Fenster *Eigenschaften von LAN-Verbindung*. Markieren Sie den Eintrag *Internetprotokoll (TCP/IP)* und klicken Sie anschließend auf *Eigenschaften*.

Wenn Ihr PC bereits in ein Netzwerk eingebunden ist, sollten Sie sich bei Ihrem Netzwerkadministrator erkundigen, ob der DHCP-Dienst unterstützt wird.

Ist das der Fall wählen Sie *IP-Adresse automatisch beziehen*.



The screenshot shows the 'Eigenschaften von Internetprotokoll (TCP/IP)' dialog box. The 'Allgemein' tab is active. In the 'IP-Adresse' section, the radio button 'IP-Adresse automatisch beziehen' is selected. In the 'DNS-Serveradresse' section, the radio button 'DNS-Serveradresse automatisch beziehen' is selected. The 'Erweitert...' button is located at the bottom right of the dialog box.

Sonst tragen Sie im folgenden Fenster IP-Adresse, Subnet-Mask und Gateway ein. Arbeitet Ihr Netzwerk mit DNS-Unterstützung, sollte auch die IP-Adresse des DNS-Servers eingetragen werden. Bestätigen Sie mit *OK*.

Damit ist die Installation von TCP/IP abgeschlossen, und Sie werden jetzt aufgefordert, Ihren PC neu zu starten.

TCP/IP-Ethernet bei gleichzeitigem DFÜ-Internetzugang

Hat der PC neben dem Zugang ins lokale Netz einen Internet-Zugang über DFÜ, bietet Windows 2000 eine Besonderheit:

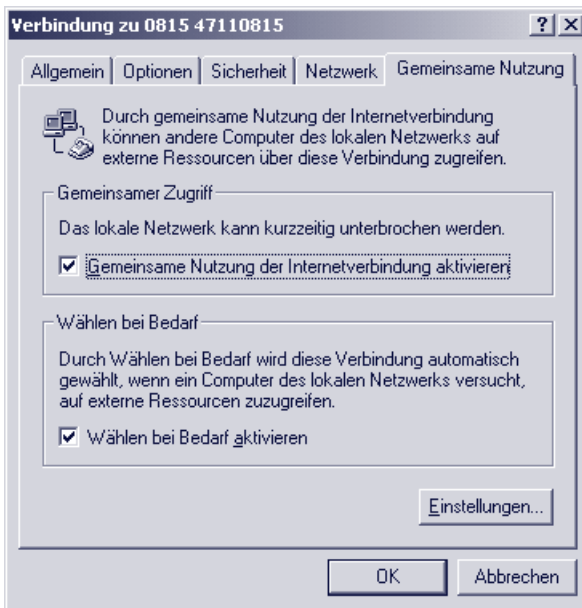


Der Internetzugang dieses PCs kann von anderen, am Netzwerkgeschlossenen Geräten mitgenutzt werden.

Um diesen Dienst freizuschalten, doppelklicken Sie in der Systemsteuerung das Icon für die angelegte DFÜ-Verbindung.



Klicken Sie auf *Eigenschaften* und wählen Sie im folgenden Fenster das Register *Gemeinsame Nutzung*



Aktivieren Sie die *Gemeinsame Nutzung der Internetverbindung*.

Der PC arbeitet nun quasi als Router ins Internet.

Durch Aktivierung der gemeinsamen Nutzung wird die IP-Adresse des PC vom System fest auf 192.168.0.1 geändert.

Darüber hinaus tritt dieser PC plötzlich als DHCP- bzw. BootP-Server auf.

Zur Erinnerung: Ein DHCP Server teilt Netzwerkteilnehmern auf Anforderung automatisch eine IP-Adresse zu. Nähere Informationen finden Sie im Kapitel DHCP

PC bei denen die *Gemeinsame Nutzung* aktiviert ist, vergeben also willkürlich IP-Adressen aus dem Adressraum 192.168.0 und das sogar auf BootP Anfragen hin.

Bei BootP ist im Normalfall nur die Vergabe reservierter Adressen vorgesehen!

Hier ist also Vorsicht geboten! Durch dieses Verhalten ist es möglich, dass andere Netzwerkteilnehmer in Folge der Adressänderung nicht mehr erreichbar sind.

Abhilfe

1. Bei kleineren Netzwerken in denen auf einem Windows 2000 PC die *Gemeinsame Nutzung* freigeschaltet ist:

- Deaktivieren Sie bei allen Netzteilnehmern das automatische Beziehen einer IP-Adresse via DHCP, bzw. BootP.
- Vergeben Sie den Netzteilnehmern festeingestellte IP-Adressen im Adressraum 192.168.0.

Durch diese Maßnahmen bleiben die Netzteilnehmer auch über Ihre IP-Adresse erreichbar. Das ist vor allem bei Embedded-Systemen wie z.B. dem Com-Server wichtig. Trotzdem lassen sich so die Vorteile der gemeinsamen Nutzung in Anspruch nehmen.

2. Bei größeren Netzen sollte auf die gemeinsame Nutzung verzichtet werden. Statt dessen empfehlen wir den Einsatz eines Routers.

Kleines Netzwerk-ABC

10Base2 – 10Mbit/s BASEband 200 (185)m/Segment

Ethernet-Topologie auf koaxialer Basis mit einer Übertragungsrate von 10MBit/s.

Weitere geläufige Bezeichnungen für 10Base2 sind auch Cheapernet oder Thin-Ethernet. Es wird Koax-Kabel mit 50 Ohm Impedanz in einer dünnen und flexiblen Ausführung verwendet, um die einzelnen Stationen busförmig miteinander zu verbinden. Anfang und Ende eines Segments müssen mit Abschlußwiderständen von 50 Ohm abgeschlossen werden.

Die Transceiver sind auf den Netzwerkkarten integriert, so dass der Bus direkt bis an jeden Arbeitsplatz geführt werden muss, wo er über BNC-T-Stücke an den Rechner angeschlossen wird. Die Dämpfung des Kabels, sowie die teilweise hohe Anzahl von Steckverbindern beschränken ein 10Base2 Segment auf max. 185m mit max. 30 Anbindungen. Zwischen zwei Stationen dürfen nicht mehr als vier Repeater liegen.

Die Schwachstelle der physikalischen Bus-Topologien von Ethernet liegt in der Tatsache, dass eine Unterbrechung des Kabels – z.B. durch Abziehen eines Steckverbinders – den Stillstand des gesamten Netzsegmentes zur Folge hat.

10Base5 – 10Mbit/s BASEband 500m/Segment

10Base5 ist die ursprüngliche Ethernet-Spezifikation. Die Verkabelung beruht hier auf einem koaxialen Buskabel mit 50 Ohm Impedanz und einer max. zulässigen Länge von 500m (Yellow Cable). Bedingt durch die koaxiale Zwei-Leiter-Technik (Seele und Schirm) lassen sowohl 10Base5 als auch 10Base2 lediglich einen Halbduplex-Betrieb zu. Die Netzwerkteilnehmer werden über externe Transceiver angeschlossen, die über Vampir-Krallen die Signale direkt vom Buskabel abgreifen, ohne dieses durch Steckverbinder o. ä. zu unterbrechen. Getrennt nach Sende-, Empfangs- und Kollisions-Information werden die Daten vom Transceiver auf einem 15-poligen D-SUB-Steckverbinder zur Verfügung gestellt. Der Anschluss des Endgerätes erfolgt über ein 8adriges

TP-Kabel von max. 50m Länge. Zwischen zwei beliebigen Stationen dürfen nicht mehr als vier Repeater liegen. Diese Regel betrifft allerdings nur „hintereinander“ liegende Repeater – bei der Realisierung baumartiger Netzwerkstrukturen kann also durchaus eine Vielzahl von Repeatern eingesetzt werden.

Durch die Verwendung von relativ hochwertigem Kabel ohne jegliche Unterbrechungen durch Steckverbinder ergeben sich die Vorteile der großen Segmentlänge und der hohen Anzahl möglicher Anbindungen pro Segment (max. 100).

Die Dicke und Unflexibilität des Yellow Cable, sowie die, durch externe Transceiver, zusätzlich entstehenden Kosten sind die Hauptnachteile von 10Base5 und haben wohl entscheidend zur Einführung von 10Base2 beigetragen.

10BaseT – 10Mbit/s BASEband Twisted Pair

Mit der Definition von 10BaseT wird die physikalische Topologie von der logischen getrennt. Die Verkabelung ist, ausgehend von einem Hub als zentraler aktiver Komponente, sternförmig ausgeführt. Es wird ein mindestens zweipaariges Kabel der Kategorie 3 mit 100 Ohm Impedanz verwendet, in dem die Daten getrennt nach Sende- und Empfangsrichtung übertragen werden. Als Steckverbinder werden 8-polige RJ45-Typen eingesetzt, in denen die Paare auf den Pins 1/2 und 3/6 aufgelegt sind. Die max. Länge eines Segments (= Verbindung vom Hub zum Endgerät) ist auf 100m begrenzt. Ihren Ursprung hat die 10BaseT-Topologie in den USA, weil sie ermöglichte, die dort üblichen Telefonverdrahtungen auch für den Netzbetrieb zu nutzen. Für Deutschland entfiel dieser Vorteil, da hier für die Telefonie Stern-4er-Kabel verlegt wurden, die den Anforderungen der Kategorie 3 nicht entsprachen.

Kabelunterbrechungen oder abgezogene Stecker, die bei allen physikalischen Busstrukturen einen Stillstand des gesamten Segmentes bedeuten, beschränken sich bei 10BaseT lediglich auf einen Arbeitsplatz.

100BaseT4 – 100Mbit/s BASEband Twisted 4 Pairs

100BaseT4 spezifiziert eine Ethernet-Übertragung mit 100Mbit/s. Wie bei 10BaseT handelt es sich um eine physikalische Sternstruktur mit einem Hub als Zentrum. Es wird ebenfalls ein Kabel der Kategorie 3 mit 100 Ohm Impedanz, RJ45 Steckverbindern

und einer max. Länge von 100m eingesetzt. Die zehnfache Übertragungsgeschwindigkeit von 100Mbit/s bei gleichzeitiger Einhaltung der Kategorie-3-Bandbreite von 25MHz wird u.a. auch durch die Verwendung aller vier Aderpaare erzielt. Für jede Datenrichtung werden bei 100BaseT4 immer 3 Paare gleichzeitig verwendet.

100BaseTX – 100Mbit/s BASEband Twisted 2 Pairs

100BaseTX spezifiziert die 100Mbit/s-Übertragung auf 2 Aderpaaren über eine mit Komponenten der Kategorie 5 realisierte Verkabelung. Kabel, RJ45-Wanddosen, Patchpanel usw. müssen gemäß dieser Kategorie für eine Übertragungsfrequenz von mindestens 100MHz ausgelegt sein.

Abschlußwiderstand

Bei koaxialen Netzwerktopologien wie 10Base5 oder 10Base2 muss jeder Netzwerkstrang am Anfang und am Ende mit einem Abschlußwiderstand (Terminator) abgeschlossen werden. Der Wert des Abschlußwiderstandes muß der Kabelimpedanz entsprechen; bei 10Base5 oder 10Base2 sind dies 50 Ohm.

Administrator

Systemverwalter, der im lokalen Netzwerk uneingeschränkte Zugriffsrechte hat und für die Verwaltung und Betreuung des Netzwerks zuständig ist. Der Administrator vergibt unter anderem die IP-Adressen in seinem Netzwerk und muss die Einmaligkeit jeder IP-Adresse gewährleisten.

ARP – Address Resolution Protocol

Über ARP wird die zu einer IP-Adresse gehörende Ethernet-Adresse eines Netzwerkteilnehmers ermittelt. Die ermittelten Zuordnungen werden auf jedem einzelnen Rechner in der ARP-Tabelle verwaltet. In Windows-Betriebssystemen kann man auf die ARP-Tabelle mit Hilfe des ARP-Befehls Einfluss nehmen.

Eigenschaften und Parameter des ARP Kommandos in der DOS-Box:

- ARP -A listet die Einträge der ARP-Tabelle auf
- ARP -S <IP-Adresse> <Ethernet-Adresse> fügt der ARP-Tabelle einen statischen Eintrag hinzu

- ARP -D <IP-Adresse> löscht einen Eintrag aus der ARP-Tabelle

ARP ist im Internet-Standard RFC-826 definiert; *vgl. a.*  26ff.

AUI – Attachment Unit Interface

Schnittstelle zur Anbindung eines externen Ethernet-Transceivers.


Getrennt nach Sende-, Empfangs- und Kollisions-Information werden die Daten vom Transceiver auf einem 15-poligen D-SUB-Steckverbinder zur Verfügung gestellt. Der Anschluß des Endgerätes erfolgt über ein 8-adriges TP-Kabel von max. 50m Länge.

Während die AUI-Schnittstelle in der Vergangenheit hauptsächlich zur Ankopplung von Endgeräten an 10Base5-Transceiver (Yellow-Cable) genutzt wurde, verwendet man sie heute eher zur Anbindung an LWL-Transceiver (Glasfaser) o.ä.

BNC – Bayonet Neill Concelmann

Bei der BNC-Steckverbindung handelt es sich um einen Bajonettverschluß zum Verbinden zweier Koaxialkabel. BNC-Steckverbindungen werden in 10Base2-Netzwerken zur mechanischen Verbindung der RG-58-Kabel (Cheapernet) verwendet.

BootP – Boot Protocol

Dieses ältere Protokoll zum Booten von PCs ohne Festplatte über das Netzwerk, ist der Vorläufer von DHCP. Auch moderne DHCP-Server unterstützen immer noch BootP-Anfragen. Heute wird BootP in erster Linie eingesetzt, um Embedded-Systemen eine IP-Adresse zuzuteilen. Dazu muss auf dem DHCP-Server ein reservierter Eintrag hinterlegt werden, in dem der MAC-Adresse des Embedded-Systems eine feste IP-Adresse zugeordnet ist; *vgl. a.*  39ff

Bridge

Bridges verbinden Teilnetze miteinander und entscheiden anhand der Ethernet-Adresse, welche Pakete die Bridge passieren dürfen und welche nicht. Die dazu notwendigen Informationen entnimmt die Bridge Tabellen, die je nach Modell vom

Administrator eingegeben werden müssen oder von der Bridge dynamisch selbst erstellt werden; *vgl. a. Router*

Broadcast

Als Broadcast bezeichnet man einen Rundruf an alle Netzteilnehmer. Eine typische Broadcast-Anwendung ist der ARP-Request (siehe ARP). Auch andere Protokolle – etwa **RIP** oder **DHCP** – nutzen Broadcast-Meldungen.

Broadcast-Meldungen werden nicht über Router oder Bridges weitergegeben.

Browser

Client-Programm mit grafischer Benutzeroberfläche, das dem Anwender die Möglichkeit gibt, Webseiten anzuzeigen und andere Dienste im Internet zu nutzen. *vgl. a. 48.*

Bus-System

Bei einem Bus-System teilen sich mehrere Endgeräte eine einzige Datenleitung (Busleitung). Da zu einer gegebenen Zeit jeweils nur ein Endgerät die Datenleitung benutzen darf, erfordern Bus-Systeme immer ein Protokoll zur Regelung der Zugriffsrechte. Klassische Bus-Systeme sind die Ethernet-Topologien 10Base2 und 10Base5.

Cheapernet

Andere Bezeichnung für Ethernet auf der Basis von 10Base2.

Client

Computer oder Anwendungen, die Dienste von sogenannten Servern in Anspruch nehmen. Server-Dienste können zum Beispiel die Bereitstellung einer COM- oder Drucker-Schnittstelle im Netzwerk, aber auch Telnet und FTP sein; *vgl. a. 19.*

Client-Server-Architektur

System der „verteilten Intelligenz“, bei dem der Client Verbindung zu einem Server aufbaut, um vom Server angebotene Dienste in Anspruch zu nehmen. Manche Server-Anwendungen können mehrere Clients gleichzeitig bedienen. *vgl. a. 19.*

Com-Server

Endgerät in TCP/IP-Ethernet Netzwerken, das Schnittstellen für serielle Geräte und digitale E/A-Punkte über das Netzwerk zur Verfügung stellt. *vgl. a.* 151ff.

DHCP – *Dynamic Host Configuration Protocol*

Dynamische Zuteilung von IP-Adressen aus einem Adressenpool.

DHCP wird benutzt, um PCs in einem TCP/IP-Netz automatisch – also ohne manuellen Eingriff – zentral und somit einheitlich zu konfigurieren. Der Systemadministrator bestimmt, wie die IP-Adressen zu vergeben sind und legt fest, über welchen Zeitraum sie vergeben werden.

DHCP ist in den Internet-Standards RFC 2131 (03/97) und RFC 2241 (11/97) definiert; *vgl. a.* 36ff.

DNS – *Domain Name Service*

Netzteilnehmer werden im Internet über numerische IP-Adressen angesprochen. Doch weil man sich Namen eben besser merken kann als Nummern, wurde der DNS eingeführt.

DNS beruht auf einem hierarchisch aufgebauten System: Jede Namensadresse wird über eine Top-Level-Domain („de“, „com“, „net“ usw.) und innerhalb dieser über eine Sub-Level-Domain identifiziert. Jede Sub-Level-Domain kann (muss aber nicht) nochmals untergeordnete Domains enthalten. Die einzelnen Teile dieser Namenshierarchie sind durch Punkte voneinander getrennt.

Wird vom Anwender zur Adressierung ein Domain-Name angegeben, erfragt der TCP/IP-Stack beim nächsten DNS-Server die zugehörige IP-Adresse.

Netzwerkressourcen sollten sinnvollerweise einen Domain-Namen erhalten, der im Kontext zu der angebotenen Dienstleistung oder dem Firmennamen des Anbieters steht. So lässt sich z.B. *wut.de* in die Top-Level-Domain *de* (= Deutschland) und die Sub-Level-Domain *wut* (= Wiesemann & Theis GmbH) auflösen; *vgl. a.* 41ff.

DNS-Server

DNS-Server stellen im Internet die Dienstleistung zur Verfügung, einen Domain-Namen in eine IP-Adresse aufzulösen.

E-Mail

Elektronische Post über Internet und Intranet; *vgl. a.* 71ff.

E-Mail-Adresse

Eine E-Mailadresse wird benötigt, um einem Anwender elektronische Post senden zu können und setzt sich immer aus dem Mailbox-Namen des Anwenders und der Ziel-Domain, getrennt durch das @-Zeichen zusammen. Ein Beispiel: *info@wut.de* bezeichnet das Info-Postfach auf dem Mailserver von W&T; *vgl. a.* 71ff.

Embedded System

Als Embedded System bezeichnet man eine mikroprozessor-gesteuerte Baugruppe, die als eingebetteter Teil eines Gerätes oder einer Maschine im Hintergrund Daten verarbeitet und ggf. Prozesse steuert.

Ethernet

Ethernet ist die zur Zeit bei lokalen Netzen am häufigsten angewandte Technologie. Es gibt drei verschiedene Ethernet Topologien – 10Base2, 10Base5 und 10BaseT –; die Übertragungsrate beträgt 10 Mbit/s; *vgl. auch* 11ff.

Ethernet-Adresse

Die unveränderbare, physikalische Adresse einer Netzwerkkomponente im Ethernet; *vgl. a.* 13.

Fast-Ethernet

Fast-Ethernet ist quasi ein Upgrade der 10BaseT-Topologie von 10MBit/s auf 100 Mbit/s. *vgl. hierzu* **100BaseT4** und **100BaseTX**

Firewall

Unter Firewall versteht man Netzwerkkomponenten, die ähnlich einem Router ein internes Netzwerk (Intranet) an ein öffentliches Netzwerk (z.B. Internet) ankoppeln. Hierbei lassen sich die Zugriffe ins jeweils andere Netz abhängig von der Zugriffsrichtung, dem benutzten Dienst sowie der Authentifizierung und Identifikation des Netzteilnehmers begrenzen oder komplett sperren.

Ein weiteres Leistungsmerkmal kann die Verschlüsselung von Daten sein, wenn z.B. das öffentliche Netz nur als Transitweg zwischen zwei räumlich getrennten Teilen eines Intranet genutzt wird.

FTP – File Transfer Protocol

FTP ist ein auf TCP/IP aufsetzendes Protokoll, das es ermöglicht, ganze Dateien zwischen zwei Netzwerkteilnehmern zu übertragen. *vgl. a.* [48] 84ff.

Gateway

Gateways verbinden – wie auch **Bridges** und **Router** – verschiedene Netze miteinander. Während Bridge und Router zwar ggf. die physikalische Art des Netzes umsetzen (z.B. Ethernet/ISDN), das eigentliche Protokoll (z.B. TCP/ IP) aber unberührt lassen, bieten Gateways die Möglichkeit, einen Zugang zu protokollfremden Netzen zu schaffen (z.B. TCP/IP auf Profibus). Ein Gateway hat also unter anderem auch die Aufgabe, unterschiedliche Kommunikationsprotokolle zu übersetzen.

Achtung: bei der Netzwerkkonfiguration in Windows-Betriebssystemen wird auch die Eingabe eines Gateways gefordert. Diese Angabe bezieht sich allerdings auf einen ggf. im Netzwerk vorhandene Router!

HTML – Hypertext-Markup-Language

Auszeichnungssprache, die über Schlüsselwörter vorgibt, wie die Inhalte im Browser angezeigt werden, wo Multimedia-Elemente zu finden sind, und welche Elemente wie verlinkt sind; *vgl. a.* [49] 52ff.

HTTP – Hypertext Transfer Protocol

Das HTTP-Protokoll setzt auf TCP auf und regelt die Anforderung und Übertragung von Webinhalten zwischen HTTP-Server und Browser. Damit ist HTTP heute das meistgenutzte Protokoll im Internet; *vgl. a.* [49] 59ff.

Hyperlink

Verweis auf andere Webseiten oder Inhalte innerhalb einer Webseite. Durch einfaches Anklicken des verlinkten Elements gelangt der Anwender auf die gewünschte Webseite. [48] 54ff.

Hub

Ein Hub – oft auch als Sternkoppler bezeichnet – bietet die Möglichkeit, mehrere Netzteilnehmer sternförmig miteinander zu verbinden. Datenpakete, die auf einem Port empfangen werden, werden gleichermaßen auf allen anderen Ports ausgegeben.

Neben Hubs für 10BaseT (10Mbit/s) und 100BaseT (100Mbit/s) gibt es sogenannte Autosensing-Hubs, die automatisch erkennen, ob das angeschlossene Endgerät mit 10 oder 100Mbit/s arbeitet. Über Autosensing-Hubs können problemlos ältere 10BaseT-Geräte in neue 100BaseT-Netzwerke eingebunden werden.

ICMP – Internet Control Message Protocol

Das ICMP-Protokoll dient der Übertragung von Statusinformationen und Fehlermeldungen zwischen IP-Netzknotten. ICMP bietet außerdem die Möglichkeit einer Echo-Anforderung; auf diese Weise läßt sich feststellen, ob ein Bestimmungsort erreichbar ist; *vgl. auch Ping*

Internet

Das Internet ist der derzeit weltweit größte Netzverbund, der den angeschlossenen Netzteilnehmern eine nahezu grenzenlose Kommunikationsinfrastruktur zur Verfügung stellt. Durch Einsatz von TCP/IP können die Netzteilnehmer plattformunabhängig im Internet angebotenen Dienste wie E-Mail, FTP, HTTP usw. in Anspruch nehmen.

Intranet

Ein abgeschlossenes Netzwerk (etwa innerhalb eines Unternehmens), in dessen Grenzen die Netzteilnehmer Internet- typische Dienste wie E-Mail, FTP, HTTP usw. in Anspruch nehmen können. In aller Regel gibt es von einem Intranet über Router bzw. Firewalls auch Übergänge in das Internet.

IP – Internet Protocol

Protokoll, das die Verbindung von Teilnehmern ermöglicht, die in unterschiedlichen Netzwerken positioniert sind.

vgl. a.  15ff.

IP-Adresse

Die IP-Adresse ist eine 32-Bit-Zahl, die jeden Netzteilnehmer im Internet bzw. Intranet eindeutig identifiziert. Sie besteht aus einem Netzwerkteil (Net-ID) und einem Benutzerteil (Host-ID).

vgl. a.  16ff.

ISDN – *Integrated Services Digital Network*

ISDN ist der neue Standard in der Fernmeldetechnik und hat das analoge Fernsprechnetz in Deutschland komplett ersetzt. Bei ISDN werden Telefon und Telefax, aber auch Bildtelefonie und Datenübermittlung integriert. Über ISDN können also abhängig von den jeweiligen Endgeräten Sprache, Texte, Grafiken und andere Daten übertragen werden.

ISDN stellt über die S0 Schnittstelle eines Basisanschlusses zwei Basiskanäle (B-Kanäle) mit je 64 kbit/s sowie einen Steuerkanal (D-Kanal) mit 16 kbit/s zur Verfügung. Der digitale Teilnehmeranschluß hat zusammengefaßt eine maximale Übertragungsgeschwindigkeit von 144 kbit/s (2B+D). In den beiden B-Kanälen können gleichzeitig zwei unterschiedliche Dienste mit einer Bitrate von 64 kbit/s über eine Leitung bedient werden.

ISDN-Router

ISDN-Router gestatten es, zwei lokale Netzwerke über das ISDN-Netz eines Telefonnetz-Providers miteinander zu verbinden. Dabei übernehmen ISDN-Router neben den normalen Funktionen eines Routers auch das Handling der ISDN-Verbindung.

LAN – *Local Area Network*

Lokales Netz innerhalb eines begrenzten Gebiets unter Anwendung eines schnellen Übertragungsmediums wie z.B. Ethernet.

MAC-ID

Die unveränderbare, physikalische Adresse einer Netzwerkkomponente (MAC = Media Access Control); vgl. a. **Ethernet-Adresse** und  13ff.

NAT – Network Address Translation

Durch die explosionsartige Ausweitung des Internet in den letzten Jahren sind freie IP-Adressen knapp geworden und werden nur noch sehr sparsam vergeben. NAT kommt dort zum Einsatz, wo Firmennetze ans Internet angebunden werden. Das Firmennetz ist über einen NAT-fähigen Router mit dem Internet verbunden, arbeitet intern allerdings mit einem eigenen vom Internet unabhängigen IP-Adressraum. Von außen ist das Netz nur über eine einzige (oder einige wenige) IP-Adresse(n) ansprechbar. Anhand der Portnummer im empfangenen TCP/IP-Paket wird dieses an einen bestimmten internen Netzteilnehmer weiter geroutet.

Ping – Packet Internet Groper

Ping dient in TCP/IP-Netzen zu Diagnosezwecken; mit Hilfe dieser Funktion lässt sich überprüfen, ob ein bestimmter Teilnehmer im Netz existiert und tatsächlich ansprechbar ist. Ping arbeitet mit dem ICMP-Protokoll, welches auf das IP-Protokoll aufsetzt. Setzt ein Netzteilnehmer durch Eingabe des Ping-Kommandos einen ICMP-Request ab, gibt die angesprochene Station einen ICMP-Reply an den Absender zurück.

Der Aufruf des Kommandos *PING <IP-Adresse>* in der DOS-Box fordert den durch die IP-Adresse angegeben Netzteilnehmer auf, eine Rückmeldung zu geben. Zusätzlich können noch diverse Parameter angegeben werden:

- t Wiederholt das Ping-Kommando in Dauerschleife, bis der Anwender mit <Strg> C unterbricht.
- n count Wiederholt das Ping-Kommando „count“ mal.
- l size „size“ gibt an, mit wieviel Byte das ICMP-Paket aufgefüllt wird. Bei Com-Servern in Default-Einstellung sind dies maximal 512 Byte.
- w timeout „timeout“ spezifiziert, wie lange (in Millisekunden) auf die Rückmeldung gewartet wird.

Ein Beispiel:

```
PING 172.16.232.49 -n 50
```

sendet 50 Ping-Kommandos an die Station 172.16.232.49.

Ist der Netzteilnehmer vorhanden, erscheint folgende Rückmeldung:

```
Reply from 172.16.232.49: bytes=32 time=10ms TTL=32
```

Bleibt die Rückmeldung aus, wird folgende Meldung zurückgegeben:

```
Request timed out.
```

Die von Ping verwendeten ICMP-Pakete sind im Internet-Standard RFC-792 definiert.

POP3 – Post Office Protocol Version 3

Um eingegangene E-Mails aus dem Postfach auf dem Mailserver abzuholen, wird in den meisten Fällen das POP3-Protokoll benutzt. Auch POP3 setzt auf TCP auf; vgl. a. 71ff, 75ff.

PPP – Point to Point Protocol

PPP ist ein erweiterter Nachfolger von SLIP und weist u.a. eine verbesserte Fehlerkorrektur auf.

Genau wie SLIP bietet PPP die Möglichkeit, TCP/IP-Geräte, die keinen LAN-Anschluß haben, über die serielle Schnittstelle in TCP/IP-Netze einzubinden.

Repeater

In lokalen Netzen dient ein Repeater zur Verbindung zweier Ethernet-Segmente, um das Netz über die Ausdehnung eines einzelnen Segmentes hinaus zu erweitern. Repeater geben Datenpakete von einem Netzwerksegment zum anderen weiter, indem sie zwar die elektrischen Signale normgerecht „auffrischen“, den Inhalt der Datenpakete dabei aber unverändert lassen. Erkennt der Repeater auf einem der angeschlossenen Segmente einen physikalischen Fehler, wird die Verbindung zu diesem Segment abgetrennt („partitioniert“). Die Partitionierung wird automatisch aufgehoben, wenn der Fehler nicht mehr vorhanden ist.

Zwischen zwei Stationen dürfen nicht mehr als vier Repeater liegen. Diese Regel betrifft allerdings lediglich „hintereinander“ liegende Repeater – bei der Realisierung baumartiger Netzwerkstrukturen kann also durchaus eine Vielzahl von Repeatern eingesetzt werden.

RIP – Routing Information Protocol

Routingprotokolle wie RIP dienen dazu, Veränderungen der Routen zwischen zwei vernetzten Systemen an die beteiligten Systeme weiterzuleiten und so eine dynamische Änderung der Routingtabellen zu ermöglichen. RIP ist im Internet-Standard RFC-1058 definiert

Router

Router verbinden zwei unterschiedliche Netze, wobei im Gegensatz zu Bridges nicht anhand der Ethernet-Adresse, sondern in Abhängigkeit von der IP-Adresse entschieden wird, welche Datenpakete weiterzuleiten sind.

vgl. a. Bridge und  22.

SLIP – Serial Line Internet Protocol

SLIP bietet eine einfache Möglichkeit zur Übertragung von TCP/IP-Datenpaketen über serielle Punkt-zu-Punkt-Verbindungen. Damit können Endgeräte, die nicht über einen LAN-Anschluß verfügen, auch über die serielle Schnittstelle ins Netzwerk eingebunden werden.


SLIP arbeitet nach einem sehr einfachen Algorithmus ohne eigene Datensicherungsverfahren: Dem eigentlichen IP-Datenpaket wird ein Startzeichen (dezimal 192) vorangestellt und ein Endzeichen (ebenfalls dezimal 192) angehängt. Um die binäre Transparenz zu erhalten, werden im Datenpaket vorkommende Start- und Endzeichen zuvor durch andere Sequenzen ersetzt. SLIP ist in RFC 1055 beschrieben.

SLIP-Router

Ein SLIP-Router stellt die Hardware und Funktionalität zur Verfügung, um serielle Endgeräte, die über einen TCP/IP-Stack verfügen, in ein Netzwerk einzubinden.

Com-Server stellen z.B. SLIP-Routing als Betriebsart zur Verfügung.

SMTP – Simple Mail Transfer Protocol

SMTP regelt den Versand von E-Mails vom Mail-Client zum Mailserver (SMTP-Server) und zwischen den Mailservern und setzt auf TCP auf; vgl. a.  71ff, 74.

SNMP – *Simple Network Management Protocol*

SNMP setzt auf UDP auf und ermöglicht die zentrale Administration und Überwachung von Netzwerkkomponenten.

SNMP ist in folgenden Standards spezifiziert: RFC 1052, RFC 1155, RFC 1156, RFC 1157, RFC 1213 und RFC 1441.

STP – *Shielded Twisted Pair*

Abgeschirmtes Datenkabel, bei dem jeweils 2 Kabeladern miteinander verdreht sind; *vgl. a. Twisted Pair*

Subnet-Mask

32-Bit-Wert, der festlegt, welcher Teil der IP-Adresse das Netzwerk und welcher den Netzwerkteilnehmer adressiert.


vgl. a.  28ff.

Switch

Ein Switch bietet wie ein Hub die Möglichkeit, mehrere Netzteilnehmer sternförmig miteinander zu verbinden. Switches vereinigen die Funktionalität eines **Hub** mit denen einer **Bridge**: Ein Switch „lernt“ die Ethernet-Adresse des, an einem Port angeschlossenen Netzteilnehmers und leitet dorthin nur noch diejenigen Datenpakete weiter, die an diesen Netzteilnehmer adressiert sind. Eine Ausnahme bilden dabei Broadcast-Meldungen, die an alle Ports weitergegeben werden (hier unterscheidet sich der Switch in seiner Funktion von einer Bridge, die Broadcast-Meldungen generell nicht weitergibt).

Neben Switches für 100Base T (100Mbit/s) gibt es sogenannte Autosensing-Switches, die automatisch erkennen, ob das angeschlossene Endgerät mit 10 oder 100Mbit/s arbeitet. Über Autosensing-Switches können problemlos ältere 10BaseT-Geräte in neue 100BaseT-Netzwerke eingebunden werden.

TCP – *Transmission Control Protocol*

TCP setzt auf IP auf und sorgt nicht nur für die Verbindung der Teilnehmer während der Datenübertragung, sondern stellt auch die Korrektheit der Daten und die richtige Abfolge der Datenpakete sicher; *vgl. a.*  19ff.

TCP/IP-Stack

Teil des Betriebssystems oder ein auf das Betriebssystem aufgesetzter Treiber, der alle für die Unterstützung des IP-Protokolls benötigten Funktionen und Treiber zu Verfügung stellt.

Telnet – *Terminal over Network*

In der Vergangenheit kam Telnet vor allem für den Fernzugriff über das Netzwerk auf UNIX-Server zum Einsatz. Über eine Telnet-Anwendung (Telnet-Client) kann von einem beliebigen Rechner im Netz ein Fernzugriff auf einen anderen Rechner (Telnet-Server) erfolgen. Heute wird Telnet auch zur Konfiguration von Netzwerkkomponenten wie z.B. Com-Servern benutzt. Telnet wird unter TCP/IP normalerweise über Portnummer 23 angesprochen; für spezielle Anwendungen können aber auch andere Portnummern verwendet werden. Telnet setzt auf TCP/IP als Übertragungs und Sicherungsprotokoll auf. *vgl. a. [8] 80ff.* Telnet ist im Internet-Standard RFC 854 definiert

TFTP – *Trivial File Transfer Protocol*

Das Trivial File Transfer Protocol (TFTP) ist neben FTP ein weiteres Protokoll zur Übertragung ganzer Dateien. TFTP bietet nur ein Minimum an Kommandos, unterstützt keine aufwendigen Sicherheitsmechanismen und benutzt UDP als Übertragungsprotokoll. Da UDP ein ungesichertes Protokoll ist, wurden in TFTP eigene minimale Sicherungsmechanismen implementiert. *vgl. a. [8] 88ff.*

Das Trivial File Transfer Protocol ist in den Standards 783, 906, 1350 und 1782 bis 1785 beschrieben.

Transceiver

Das Wort Transceiver ist eine Zusammensetzung aus Transmitter (Sender) und Receiver (Empfänger). Der Transceiver realisiert den physikalischen Netzzugang einer Station an das Ethernet und ist bei den modernen Ethernet-Topologien 10Base2 und 10BaseT auf der Netzwerkkarte integriert. Nur bei 10Base5 (*vgl. auch AUI-Anschluß*) ist der Transceiver als externe Komponente direkt am Netzwerkkabel angebracht.

Twisted Pair

Datenkabel, bei dem jeweils zwei Kabeladern miteinander verdreht sind. Durch die paarige Verseilung einzelner Doppeladern wird ein deutlich reduziertes Übersprechverhalten zwischen den Doppeladern in einem Kabel erreicht. Man unterscheidet bei Twisted-Pair-Kabeln zwischen ungeschirmten UTP-Kabeln (Unshielded Twisted Pair) und geschirmten STP-Kabeln (Shielded Twisted Pair).

TP-Kabel werden vor allem in der Netzwerktechnik eingesetzt und sind nach ihren maximalen Übertragungsfrequenzen kategorisiert; in der Praxis kommen heute meist zwei Typen zum Einsatz:

- Kategorie-3-Kabel erlauben eine maximale Übertragungsfrequenz von 25MHz, ausreichend für den Einsatz in 10BaseT-, aber auch 100BaseT4-Netzen.
- Kategorie-5-Kabel erlauben eine maximale Übertragungsfrequenz von 100MHz und reichen damit für alle heutigen Netzwerktopologien aus.

UDP – User Datagram Protocol

UDP ist ein Protokoll, das wie TCP auf IP aufsetzt, im Gegensatz dazu aber verbindungslos arbeitet und über keine Sicherheitsmechanismen verfügt. Der Vorteil von UDP gegenüber IP ist die höhere Übertragungsgeschwindigkeit. *vgl. a. [22]*.

URL – Uniform Resource Locator

Adress- und Protokollinformation für den Browser. Über den URL gibt der Anwender dem Browser vor, welches Protokoll genutzt wird, auf welchem Webserver die Seite liegt, und wo diese auf dem Webserver zu finden ist. *vgl. a. [49f]*.

UTP – Unshielded Twisted Pair

Im Gegensatz zu **Twisted Pair** ein nicht abgeschirmtes Datenkabel, bei dem jeweils zwei Kabeladern miteinander verdreht sind.

Web-Based Management

Unter Web-Based Management versteht man die Möglichkeit, ohne spezielle Software Endgeräte übers Netzwerk direkt im Browserfenster zu konfigurieren.

WWW – World Wide Web

WWW wird häufig mit dem Internet gleichgesetzt. Das stimmt nicht ganz: Während das Internet die physikalischen Verbindungswege beschreibt, definiert das WWW einen Standard, der dem Anwender über eine grafische Benutzeroberfläche durch einfachste Bedienung Zugang zu den gängigsten Internetdiensten verschafft. Per Mausklick lassen sich Webseiten anfordern, E-Mails verschicken und Dateien downloaden. *vgl. a. 48ff.*

Zahlensysteme

Neben dem dezimalen Zahlensystem (Zeichenvorrat: 0–9, neue Stelle bei 10) werden in der Computertechnik auch oft das binäre Zahlensystem (Zeichenvorrat 0–1, Stellensprung bei 2) und das hexadezimale Zahlensystem (Zeichenvorrat: 0–9 + A–F, neue Stelle bei 16) verwendet.
In der folgenden Tabelle finden Sie einige Beispiele für die Darstellung gebräuchlicher Werte in den drei Zahlensystemen:

| Binär | Dez. | Hex. | Binär | Dez. | Hex. |
|-------|------|------|----------|------|------|
| 0 | 0 | 0 | 11111 | 31 | 1F |
| 1 | 1 | 1 | 100000 | 32 | 20 |
| 10 | 2 | 2 | ... | ... | ... |
| 11 | 3 | 3 | 111111 | 63 | 3F |
| 100 | 4 | 4 | 1000000 | 64 | 40 |
| 101 | 5 | 5 | ... | ... | ... |
| 110 | 6 | 6 | | | |
| 111 | 7 | 7 | ... | ... | ... |
| 1000 | 8 | 8 | 1111111 | 127 | 7F |
| 1001 | 9 | 9 | 10000000 | 128 | 80 |
| 1010 | 10 | A | 11000000 | 192 | C0 |
| 1011 | 11 | B | 11100000 | 224 | E0 |
| 1100 | 12 | C | 11110000 | 240 | F0 |
| 1101 | 13 | D | 11111000 | 248 | F8 |
| 1110 | 14 | E | 11111100 | 252 | FC |
| 1111 | 15 | F | 11111110 | 254 | FE |
| 10000 | 16 | 10 | 11111111 | 255 | FF |

Web-IO

Der wohl entscheidendste Grund, ein Gerät mit einer Netzwerkschnittstelle auszurüsten, war in der Vergangenheit die hohe Übertragungsgeschwindigkeit. Mit der immer größeren Ausdehnung von Firmennetzen und dem Zusammenwachsen von Intranet und Internet, gewinnt die große Standortflexibilität und die Benutzung von vorhandener Infrastruktur immer mehr Gewicht bei der Entscheidung, nicht nur PCs, File-Server und Drucker mit einem Netzwerkanschluss auszurüsten.

Zum Abschluss möchten wir Ihnen die Idee vorstellen verschiedenste Signale direkt über Netzwerk zu erfassen, zu steuern und auszuwerten.

Wir haben für diese Technik den Namen Web-IO gewählt.

Com-Server - Anwendungsbeispiele aus der Praxis

Com-Server, das sind kleine Boxen die mit einem Ethernet-Anschluss auf der einen und mit bis zu 4 seriellen Ports auf der anderen Seite ausgerüstet sind.

Der Ethernet-Anschluss arbeitet je nach Modell mit 10Mbit oder mit 10/100Mbit autosensing Technik (automatische Erkennung). Für die serielle Seite kann RS232, RS422, RS485 oder 20mA gewählt werden.

Die Unterstützung der Protokolle TCP, FTP und Telnet (jeweils als Client und Server) sowie UDP, SNMP, BootP, RARP und ARP erlaubt nahezu jede denkbare Applikation.

Als Spannungsversorgung sind 230V, 12-24V, 5V (TTL-kompatibel) und 3V möglich. Ausführliche Datenblätter der verschiedenen Com-Server finden Sie im Anhang.

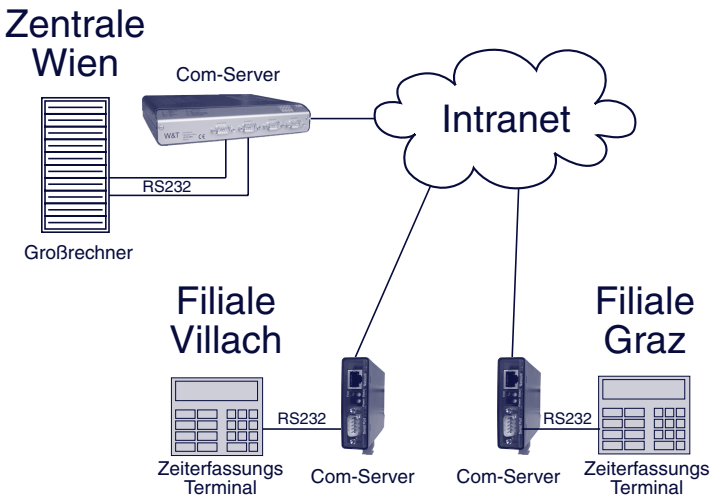
Box-to-Box - Der Tunnel durchs Netzwerk

Zwei Com-Server werden so konfiguriert, dass Daten, die am seriellen Port des 1. Com-Servers eingehen, über das Netzwerk zum 2. Com-Server weitergeroutet werden. Der 2. Com-Server gibt die seriellen Daten dann wieder aus. Das Ganze funktioniert selbstverständlich in beide Richtungen.

Ein Beispiel:

Die Zeiterfassungsdaten bei einer Bank in Wien werden über RS232 vom Zeiterfassungs-Terminal an einen UNIX-Großrechner übertragen. Die Daten aus den Zweigstellen Villach und Graz wurden bislang auf Diskette gezogen und mit der Post verschickt.

Heute werden die Daten über Com-Server im Box-to-Box Modus einfach über die ohnehin bestehende Intranet Verbindung mit übertragen. Der erste Com-Server „steckt“ die RS232-Daten in TCP-Pakete und routet sie durchs Netz an den zweiten Com-Server. Der „packt“ die RS232-Daten wieder aus und gibt sie an den Zentral-Rechner.



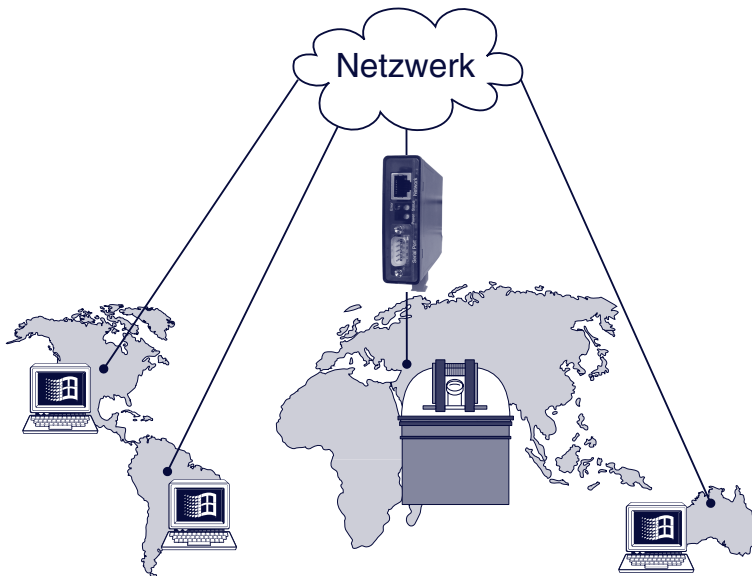
Die COM-Umlenkung - Der „ganz wo anders“ COM-Port

Mit Hilfe des Com-Umlenkungs Treibers und eines oder mehrerer Com-Server lassen sich in Microsoft Windows Betriebssystemen zusätzliche COM-Ports einrichten, die an einer beliebigen Position im Netzwerk sein können.

Ein Beispiel:

Ein Sonnenteleskop in Südeuropa übergibt seine Bilddaten über ein TCP/IP-Netzwerk an verschiedene Universitäten auf der ganzen Welt. Die Positionierungskoordinaten des Teleskops aber lassen sich leider nur über eine RS232-Schnittstelle, direkt vor Ort eingeben. Bisher mussten diese Parameter telefonisch einem Mitarbeiter durchgegeben werden, der die nötigen Einstellungen vorgenommen hat.

Seit der Konfigurationsport des Sonnenteleskops an einen Com-Server angeschlossen ist, können die Benutzer in Sydney, Washington und Tegucigalpa über die nun virtuelle COM3 ihres PC das Sonnenteleskop online positionieren.

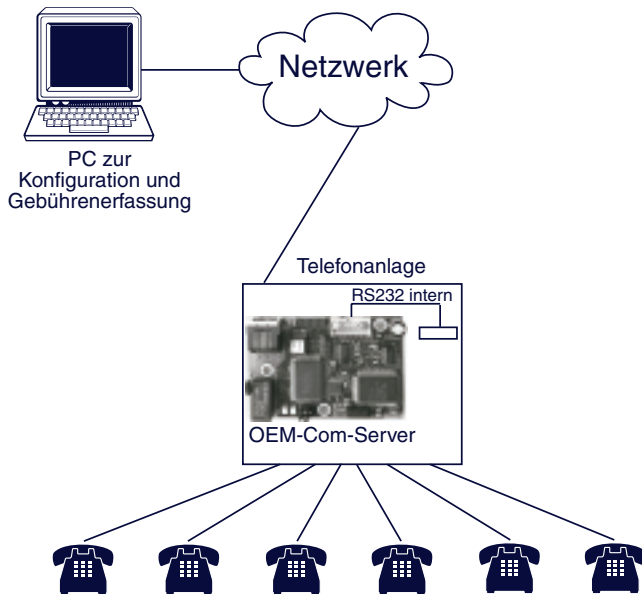


TCP/IP-Sockets - Mit dem eigenen Programm auf den seriellen Port

Über TCP oder auch UDP erlaubt der Com-Server eine direkte Kommunikation mit dem seriellen Port des Com-Servers.

Ein Beispiel:

Ein Hersteller von Telefonnebenstellenanlagen schließt die RS232 Konfigurations- und Gebührendatenschnittstelle an die eingebaute Com-Server-OEM Platinen an. Um die Gebühren-
daten zu erfassen und die Anlage zu konfigurieren wurde eine kleine Software programmiert, die das bequem übers Netzwerk erledigt.

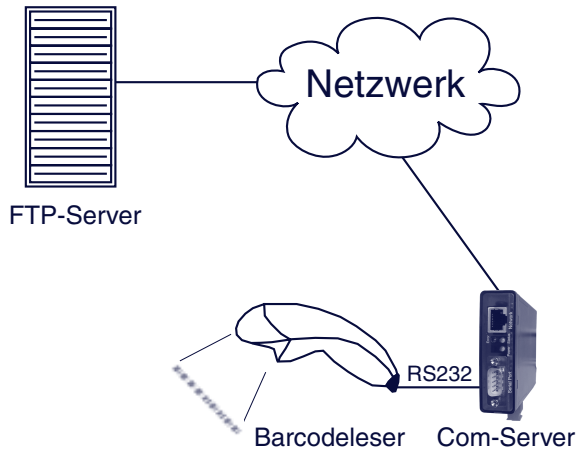


FTP - Serielle Daten direkt in eine Datei

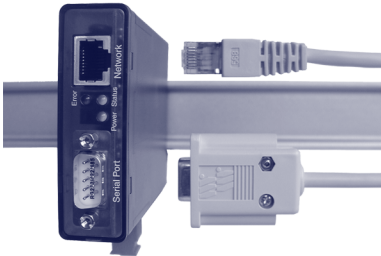
Der Com-Server unterstützt unter anderem auch FTP, als Client oder Server. Damit ist ein Dateitransfer zum oder vom seriellen Port des Com-Server problemlos möglich.

Ein Beispiel:

Im Lager einer Spedition sollen alle ein- und ausgehenden Pakete per Barcode erfasst werden. Dazu wurden für Ein- und Ausgang je ein Barcode-Leser mit einem Com-Server verbunden, der als FTP-Client konfiguriert wurde. Die eingelesenen Barcodes werden nun automatisch per FTP in Dateien auf dem File-Server der Spedition geschrieben.



Natürlich gibt es eine Vielzahl weiterer Anwendungsbeispiele für den Einsatz von Com-Servern. CNC, DNC, Betriebsdatenerfassung, Messwerterfassung, Fernwartung..... um hier nur einige zu nennen.

Com-Server - Die verschiedenen Modelle**Com-Server Highspeed Industry - #58631**

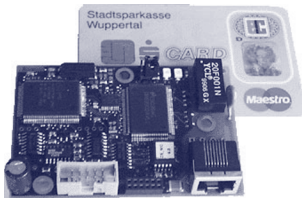
- Netzwerk: 10BaseT oder 10/100BaseT autosensing
- Protokolle TCP/IP:
UDP/TCP-Sockets, FTP, Telnet je Client und Server,
Virtueller COM-Port mit Windows COM-Umlenkung,
Box-to-Box Modus (RS232-Tunnel),
Hilfsprotokolle: ARP, RARP, DHCP/BOOTP, PING, RIP, SNMP,
HTTP und Web-Based Management (in Vorbereitung),
Inventarisierung, Gruppenmanagement
- Serieller Port: DB9-Stecker mit RS232 PC-Belegung,
umschaltbar auf RS422, RS485,
Baudrate: 50 - 230.400 Baud,
Datenformat: 7,8
Datenbit: 1,2 Stopbit,
Parity: none, even, odd,
Handshake: Hardware, Xon/Xoff
- Stromversorgung: 12-24V AC/DC auf Klemme für Industrie-
einsatz und 230V-Steckernetzteil für Büroanwendungen
- schmales Hutschienengehäuse 105x75x22 mm

Com-Server Highspeed - #58031, 58034



- Netzwerk und Protokolle:
wie Com-Server Highspeed Industry
- 1 oder 4 Serielle Ports: DB9-Stecker mit RS232 PC-Belegung,
einzeln umschaltbar auf RS422, RS485,
20mA optional möglich,
Baudrate: 50 - 230.400 Baud,
Datenformat: 7,8 Datenbit, 1,2 Stopbit,
Parity: none, even, odd, Handshake: Hardware, Xon/Xoff
- Stromversorgung: eingebautes 230V-Netzteil
- Aluminium-Tischgehäuse 212x168x40 mm

OEM Platinen



- Netzwerk: 10BaseT, oder 10/100BaseT autosensing
RJ45, Schneidklemme oder Stiftleiste möglich
- Seriell: Stiftleisten mit RS-232-TTL-Signalen oder RS485,
RS422 oder 20mA optional möglich,
- Stromversorgung: 3V, 5V oder 24V
- verschiedene Formate

Weitere Bauformen siehe <http://www.wut.de>

Web-IO - Anschlussbeispiele aus der Praxis

Web-IO das sind kleine Baugruppen, mit denen analoge und digitale Signale per TCP/IP-Ethernet gesteuert und überwacht werden können.

Ein integrierter HTTP-Server erlaubt ein vollständiges Web-Based Management. Konfiguration, Steuerung und Auswertung sind ohne gerätespezifische Spezialsoftware, selbst für den unbedarften Anwender, sofort von jedem Browser möglich.

Zusätzlich ist die Einbindung in bestehende Management- und Visualisierungssysteme ebenfalls kein Problem. SNMP und OPC, aber auch der direkte Zugang via TCP und UDP machen eine Integration ganz einfach.

SNMP-Traps und Email-Versand, bei vorhandener Infrastruktur bis hin zur SMS, erlauben aber auch ganz neue Wege der Signalverarbeitung.

Netzwerkseitig sind alle Web-IO mit einem 10/100BaseT auto-sensing Anschluss ausgestattet. Die Spannungsversorgung ist in einem weiten Bereich zwischen 12 V und 24 V Gleich- oder Wechselspannung bzw. über ein 230 V Netzteil möglich.

Durch diese Flexibilität bietet Web-IO alle Eigenschaften für den Einsatz in Anlagen- und Haustechnik, sowie Labor- und Büroanwendungen.

Web-IO Thermometer - Temperaturüberwachung im Netz

Das Web-IO Thermometer erlaubt den Anschluss von bis zu acht Temperatursensoren in NTC oder PT100 Technik. Die Temperaturen lassen sich jederzeit im Browser abrufen; als Tabelle oder in einer selbst erstellten Homepage. Für jeden Sensor können individuelle Grenzwerte festgelegt werden. Bei Grenzwertüber- oder unterschreitung kann per E-Mail oder SNMP-Trap Alarm geschlagen werden.

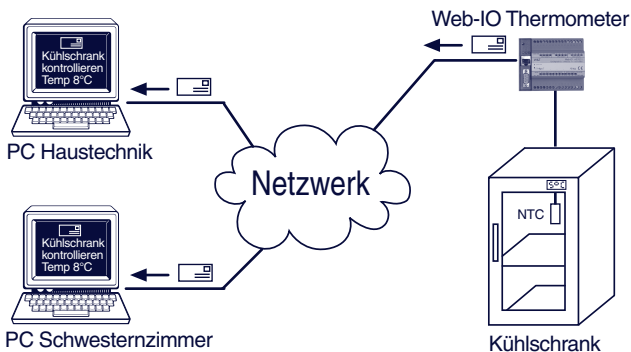
Ein Beispiel:

In einem Krankenhaus müssen spezielle Medikamente in einem Kühlschrank zwischen 3° und 8° C gelagert werden.

In der Vergangenheit wurde die Temperatur des Kühlschranks einmal pro Stunde von der Stationsschwester kontrolliert und in eine Tabelle eingetragen.

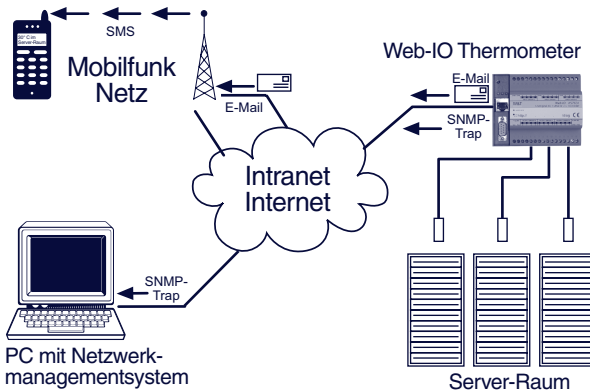
Heute überwacht das Web-IO Thermometer den Kühlschrank. Steigt die Temperatur über 7,5° C bekommt die Stationsschwester eine E-Mail. Übersteigt die Temperatur sogar 8° C wird zusätzlich eine E-Mail an die Haustechnik gesendet.

Zusätzlich wird einmal pro Woche der gesamte Temperaturverlauf per Download als Excel-Tabelle gesichert.



Ein zweites Beispiel:

In einem Rechenzentrum sind bereits mehrmals diverse Festplatten den „Hitzetod“ gestorben, weil über Nacht die Klimaanlage des Serverraumes ausgefallen ist. Heute sendet das eingesetzte Web-IO Thermometer, via E-Mail an den Mobilfunknetzbetreiber, eine SMS direkt an den Techniker. Zusätzlich wird ein SNMP-Trap an das Netzwerkmanagementsystem gesendet. So kann zu jeder Uhrzeit rechtzeitig reagiert werden.



Web-IO 12xDigital

Über HTTP (Browser), TCP, UDP SNMP oder OPC können 12 digitale Eingänge und Ausgänge per Netzwerk gesteuert und ausgewertet werden. Zusätzlich steht ein Box-to-Box Modus zur Verfügung, mit dem über einen Eingang an Web-IO 1 ein Ausgang an Web-IO 2 gesetzt werden kann.

Die digitalen Eingänge sind zu Gruppen à 4 Stück galvanisch getrennt und lassen sich mit ± 30 V ansteuern.

Die Ausgänge schalten über einen gemeinsamen Versorgungseingang Spannungen von 6 bis 30 V. Der maximale Ausgangsstrom pro Signal beträgt 500mA. Ein thermischer Überlastschutz sorgt dabei für Kurzschlussfestigkeit. Die Ausgänge können paarweise oder in Gruppen von je 4 Ausgängen parallelgeschaltet werden, um auch höhere Schaltströme zu realisieren. Freilaufdioden für Relaisanschaltung sind natürlich integriert.

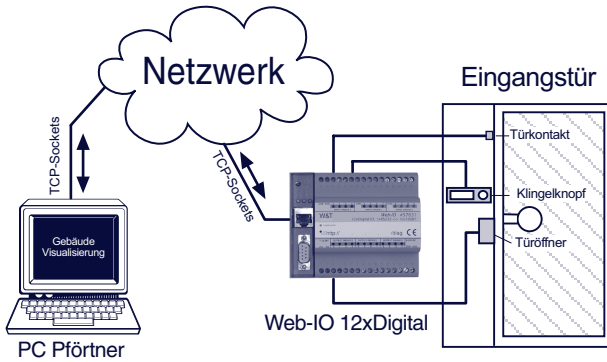
Über 12 unabhängige Alarmkonfigurationen lassen sich beliebige Eingangsmuster überwachen und im Alarmfall kann per E-Mail, SNMP-Trap oder UDP-Meldung gewarnt werden.

Auf den Punkt gebracht: mit Web-IO 12xDigital kann jedes Gerät, das einen Kontakt oder ein elektrisches Signal zur Verfügung stellt, ins Netzwerk gebracht werden.

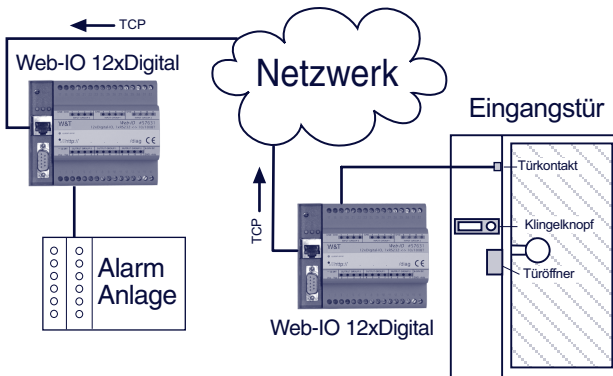
Ein Beispiel:

Der Hintereingang eines Firmengebäudes hat einen Klingelknopf, einen Türöffner und einen Kontakt, der überwacht, ob die Tür geschlossen ist.

Klingelknopf und Türkontakt sind über die Eingänge 0 und 1 an den Web-IO angeschlossen. Ausgang 0 des Web-IO steuert den Türöffner. Über TCP-Sockets sind diese Signale des Web-IO in ein Gebäudevisualisierungssystem eingebunden. So bekommt der Pförtner am Haupteingang an seinem PC ein akustisches Signal, wenn jemand den Klingelknopf drückt. Durch Mausklick kann er den Türöffner betätigen. Ob die Tür offen oder geschlossen ist wird ebenfalls angezeigt.

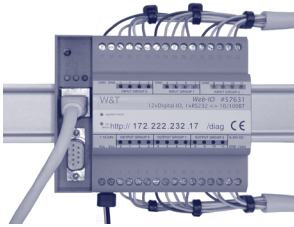


Zusätzlich ist der Türkontakt an Eingang 4 des Web-IO angeschlossen. Im Keller, neben der Alarmanlage, befindet sich ein zweiter Web-IO. Ausgang 4 dieses Web-IO steuert einen Alarmeingang der Alarmanlage an. Die Web-IO wurden so konfiguriert, dass Eingang 4 des ersten Web-IO Box-to-Box mit Ausgang 4 des zweiten Web-IO zusammenarbeitet. Der Zustand des Türkontaktes ist auf diese Weise über das Netzwerk 1:1 mit der Alarmanlage verbunden. Wird in der Nacht, wenn die Alarmanlage scharf geschaltet ist, die Tür geöffnet, schlägt diese sofort an.



Web-IO - verschiedene Modelle**Web-IO Thermometer #57603, 57604**

- Netzwerk: 10BaseT oder 10/100BaseT autosensing
- Protokolle TCP/IP:
HTTP und Web-Based Management,
Upload von anwenderspezifischen Web-Seiten,
Java-Applets, Download von Excel-Tabellen im CSV-Format,
UDP/TCP-Sockets, SMTP (Alarm per E-Mail),
SNMP (auch Trap-Funktion),
Hilfsprotokolle: ARP, RARP, DHCP/BOOTP, PING,
Inventarisierung, Gruppenmanagement
- Synchronisation von Datum und Uhrzeit mit Time-Server
- 2 Sensoreingänge #57603 bzw. 8 Sensoreingänge #57604
NTC oder PT100 (2,3,4 Leiter) möglich
- Messbereich: -45°C ... $+75^{\circ}\text{C}$
- Auflösung: $1/10^{\circ}\text{C}$
- Messfehler: $\pm 0,3^{\circ}\text{C}$, $\pm 5\%$ bei NTC, $\pm 0,3^{\circ}\text{C}$ $\pm 2\%$ bei PT100
Speicherfrequenz: 1, 5, 15, 60 Messungen/min.
Speichertiefe: 64KByte bei 2Byte pro Messwert
- Serieller Hilfsport: DB9-Stecker mit RS232 PC-Belegung
- Stromversorgung: 12-24V AC/DC auf Klemme für Industrie-
einsatz und 230V-Steckernetzteil für Büroanwendungen
- Hutschienengehäuse 107x88x63 mm

Web-IO 12xDigital #57630, 58631

- Netzwerk: 10BaseT oder 10/100BaseT autosensing
- Protokolle TCP/IP:
 - HTTP und Web-Based Management,
 - Upload von Anwenderspezifischen Web-Seiten
 - UDP/TCP-Sockets je Client und Server
 - SMTP (Alarm per E-Mail),
 - SNMP (auch Trap-Funktion),
 - Box-to-Box Betrieb zum Schalten zwischen 2 Standorten
 - Hilfsprotokolle: ARP, RARP, DHCP/BOOTP, PING,
 - Inventarisierung, Gruppenmanagement
- Synchronisation von Datum und Uhrzeit mit Time-Server
- 12 digitale Eingänge
 - in 3 Gruppen mit 2kV galvanisch getrennt
 - Eingangsspannungsbereich 30V
 - (unter 5V = Off über 12V = On)
 - 32 Bit Zähler für jeden Eingang
- 12 digitale Ausgänge
 - Schaltspannung 6 - 30V
 - Max. Schaltstrom 500mA pro Ausgang
 - kurzschlussfest durch thermische Sicherung
 - Zusammenschalten von Ausgängen für höhere Lasten möglich
- Serieller Port: DB9-Stecker mit RS232 PC-Belegung
 - bei #57630 Hilfsport zur Konfiguration
 - bei #57631 serielle Com-Server Funktion wie #58631
 - Baudrate: 50 - 230.400 Baud
 - Datenformat: 7,8 Datenbit, 1,2 Stopbit,
 - Parity: none, even, odd, Handshake: Hardware, Xon/Xoff
- Stromversorgung: 12-24V AC/DC auf Klemme für Industrie-einsatz und 230V-Steckernetzteil für Büroanwendungen
- Hutschienengehäuse 107x88x63 mm

Weitere Infos

Wir arbeiten ständig daran Ihre und unsere Ideen in neue und verbesserte Produkte umzusetzen.

Wenn sie möchten, dass wir Sie im Bereich Web-IO auf dem Laufenden halten, werden Sie einfach Mitglied im Web-IO Club!

Das geht ganz einfach:

Auf unserer Webseite

<http://www.wut.de>

im Bereich *Service* auf den Link *Info-Clubs* klicken und das Formular ausfüllen.

Sie bekommen dann von uns alle 4 - 6 Wochen eine kurze E-Mail mit den aktuellsten Informationen.

Natürlich halten wir auch auf unserer Webseite tagesaktuell Informationen zu Web-IO, Netzwerktechnik und seriellen Schnittstellen für Sie bereit.



W&T

www.wut.de

Wiesemann & Theis GmbH

Produkte und Interfaces für:

Netzwerk

Seriell

Parallel

USB

LWL

und vieles mehr

www.wut.de



Technische Grundlagen: konkret, kompakt, verständlich

Ethernet, TCP/IP und Web-IO verstehen

Einen Toaster steckt man in die Steckdose, ohne zu wissen woher der Strom kommt. Einen Videorecorder zu programmieren, ohne etwas über Sendekanäle oder Showview zu wissen, ist bekanntlich tückisch.

Wer Geräte an Computernetze anschließen will, wird es ohne Grundkenntnisse der Netzwerktechnik nicht weit bringen; insbesondere dann, wenn es sich um größere professionelle Netze handelt. Aber auch bei der Kinderzimmervernetzung kann die Kenntnis der Grundbegriffe einigen Ärger sparen.

Dicke Bücher über Netzwerktechnik finden sie genug. In diesem Büchlein sind die wesentlichen Grundlagen übersichtlich zusammengefasst. Alles was nur den Entwickler von Netzwerkprodukten interessieren muss, haben wir konsequent weglassen. Dafür sind die Informationen, die auch den Anwender interessieren könnten, durchaus mit technischem Tiefgang behandelt.



Wiesemann und Theis GmbH
Wittener Straße 312
D-42279 Wuppertal

Mail info@wut.de
Web www.wut.de

Tel. 0202 26 80-110
Fax 0202 26 80-265